2017

Modeling Mortgage Assessment with Computational Argumentation Theory and Defeasible Reasoning

Henrik Szucs
Technological University Dublin

Recommended Citation

Szucs, H. (2017) Modeling mortgage assessment with computational argumentation theory and defeasible reasoning, Dissertation, Technological University Dublin.

Modeling mortgage assessment with computational argumentation theory and defeasible reasoning

Student Name: Henrik Szucs

A dissertation submitted in partial fulfilment of the requirements of

Dublin Institute of Technology for the degree of

M.Sc. in Computing (Stream: Advanced Software Development)

Date: 2017

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Advanced Software Development Stream), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institutes guidelines for ethics in research.

Signed:

Date:

Abstract

In the mortgage lending business of a bank, a key focus area is risk analysis, which supports the mortgage awarding process and the prediction of the risk of defaulting (repayment issues). The standard risk assessment method at most banks is a scorecard calculation. A new way of predicting the defaulting is proposed, which has not been done before, using Defeasible Reasoning (DR) and computational Argumentation Theory (AT), which are areas of interdisciplinary research, in the discipline of Artificial Intelligence (AI). Argumentation is formalised by reasoning models which are inspired by human reasoning. For a more realistic representation AT employs DR which is a non-monotonic reasoning process, meaning that in case of new evidence a previous conclusion might change. The computational AT approach is predominantly knowledge driven and it includes building and evaluating arguments, their relationships, the resolution of their inconsistencies and the generation of defeasible conclusions, on which the experiment conducted in this thesis is based upon. It is demonstrated how is it possible to reasoning on a defeasible way to predict the risk of defaulting. Results demonstrated that in 75% of the test cases AT predicted better. The prediction accuracy was compared using a so-called confusion-matrix (2 by 2 squares) where the two axes being Predicted (Yes; No) and Actual (Yes;No) and a limitation applied as only half of the confusion-matrix could be used, due to unavailable and unverifiable data.

Keywords: Defeasible Reasoning, Computational Argumentation Theory, Mortgage Assessment, Risk of Defaulting

Acknowledgments

Many thanks to my supervisor Luca Longo for his teaching, his great support and guidance.

Sincere thanks to my co-supervisor Lucas Rizzo for his continuous help and aid.

I can not thank enough to all my other teachers, who shared their knowledge and highlighted the path of wisdom, especially to Damian T. Gordon for the overwhelming experiences in his lectures, a great example to follow.

Special thanks to Colin Kane, my hiring manager at work at the time, who showed his way of progressing in life, by signing up himself to study for the M.Sc. in Computing in DIT and encouraging his entire team to do the same.

Also not to forget, big thank you to all my classmates, we supported each other both in and outside class. This achievement would not have been possible without you.

Thank you to Maire-Ide Maguire, a native English speaker, who reviewed the grammar of the latest draft of the thesis.

Contents

List of Figures

List of Tables

List of Acronyms

AT Argumentation Theory

BTL Buy To Let

CBI Central Bank of Ireland

DR Defeasible Reasoning

ICB Irish Credit Bureau

ICT Information and Communications Technology

ILO Income LeftOver

LGD Loss Given Default

LTI Loan To Income

LTV Loan To Value

MILO Minimum Income LeftOver

NDI Net Disposable Income

PD Probability of Defaulting

PDH Primary Dwelling Home

Chapter 1

Introduction

1.1 Background

In the banking industry the most profitable products are the mortgages, but they are also the riskiest, because of the relative large time lines from when mortgage loans are awarded to the applicants, as they will pay it back over a long period of time. The average mortgage loan term at the analysed medium sized Irish bank is 27.1 years. During this period many conditions could change in the applicants circumstances. There are also personal conditions, like health, social and financial circumstances, which could and will change during the several decades of the life of the loan and there are financial market influencers which are measured by macro economic indicators, like Gross Domestic Product (GDP), Gross National Product (GNP), unemployment rate, minimum hourly payment, average national salary, etc. Beside normal business risk in the finance industry, the global financial crisis between 2008 and 2012 heightened the importance of risk assessment and opened up the space for researching other methods than the banking industry's standard, which is the so-called Scorecard method. That method which gives the most accurate results for predicting the risk of defaulting would be the most advantageous and most desired. Currently the accuracy of defaulting predictions, the mortgage applicants who will run into re-payment difficulties is around 66%, so 1 in 3 cases do not predict well, however, strictly speaking, the long term arrears cases represent only 10%, which brings the prediction accuracy up to 90%. An important factor is if the mortgage acceptance is too strict, then the banks are not accepting many possibly well performing cases which could have generated additional income and profit.

1.2 Research Project/Problem

Banks to avoid loosing money must perform their risk assessment as accurately as possible. Technology improvements in the past two decades have made it possible and affordable to employ personal computer hardware with special statistical software to do advanced calculations on risk assessment. In current times the outcome of the mortgage risk assessment is called the Probability of Defaulting (PD), which is a numeric value on a scale, where the two opposite ends of the scale represent Low Risk and High Risk. Defaulting means difficulties in repaying the loan, can be temporary or permanent. The method to calculate the PD in use is most commonly the Scorecard method, which is a formula driven algorithm based calculation. The execution of the Scorecard method is usually automated with a software. An entire industry is specialised on finance software where basic statistical methods were a good start in the past, but as technology evolved, the requirements increased. The banking industry is shifting to a more scientific approach and employing several subdivisions of the Artificial Intelligence (AI) discipline, like Machine Learning (ML) or Artificial Neural Networks (ANN), which leads to the research problem:

Q: Is it possible to use Defeasible Reasoning (DR) and computational Argumentation Theory (AT) to successfully model and measure the risk of mortgages?

1.3 Research Objectives

The objectives of the thesis are:

- **Obj. 1. - To review existing literature.** The foundation of all secondary research is to study what other researchers and specialists in the relevant field of finance, AT and DR have published, what are the already investigated new phenomena. The researcher's results can be obtained in the academic publications and sometimes in patents. Theoretical knowledge has to be gathered to be able to formulate the details of how the experiment what could be designed, executed, verified and assessed.

- **Obj. 2. - Designing a solution for the AT approach.** Learning from the availabilities in the literature review, the most appropriate theories need to be selected and interpreted on a way which can be applied to streamline the process. Designing the AT approach via the set of arguments, the conflicts and propose a way how to resolve the conflicts and get to a conclusion. This is the first practical phase of the experiment, but still at the theoretical stage.

- **Obj. 3. - To execute the experiment.** Has to happen in a controlled manner according to the design paying attention to the details if an iteration is required then only one detail should be changed at a time, so that the different results of the different setups can be measured, obtained and compared and if still needed then iterated further.

- **Obj. 4. - Evaluation.** The experiment and it's results need to be evaluated, for both the bank's risk prediction and the computational AT approach, ensuring that the way they are measured is the same.

- **Obj. 5. - To interpret the findings.** Categorized to 8 use cases, separated by three factors, the so-called semantics, the categories of defaulting (irrecoverable only or with ever defaulted), and the resolution of no conflict free cases.

- **Obj. 6. - Definition of contribution.** Impact on general human knowledge respective to the summary of the findings, the possibility of modeling risk with AT and DR are assessed and confirmed.

These objectives are the guiding principles for every phase of the design, implementation and analysis.

1.4 Research Methodologies

This study analyses the available academic literature as a **secondary study**, based on works of other researchers and academics, who provide state of the art solutions to the problem. The experiment is a computational Argumentation Theory (AT) approach for implementing Defeasible Reasoning (DR) in practice, where both the input parameters and output results can be quantified, thus a **quantitative** research methodology is adopted. To resolve the research problem in a **constructive** method, beside the bank's method as a benchmark, using a new alternative solution to resolve an existing problem with DR and computational AT. The study sets the scene by describing the environment and the design details of the experiment, and then continues on a **deductive** way, from the general problem to the specific details.

TYPE	**OBJECTIVE**	**FORM**	**REASONING**
Primary	Qualitative	Exploratory	***Deductive***
Secondary	***Quantitative***	***Constructive***	Inductive
	Mixed	Empirical	

Table 1.1: Research methodologies categories

1.5 Scope and Limitations

The scope of this research study on a theoretical level is derived from the Artificial Intelligence (AI) discipline to research the possibility of measuring risk assessment of mortgages with AT and DR, at a medium size Irish bank. The bank provided a data set which contained sixty-eight thousand records (rows), where each record represented a mortgage application and captured the important features about the applicants, the contracts and also contained a numeric score which indicated the risk of defaulting. A low score indicated low risk and a high score indicated high risk. The main limitation of the data set itself was, that it only contained accepted mortgage applications, only those mortgages which were drawn down. Hence it was not possible to compare cases which represented high risk in the data set. From the business point of view the the high risk cases are believed to be those applicants who would have had difficulties in repaying the mortgage. In reality that does not meanthat those cases would have had performed badly, but this can never be verified. These unjustifiable cases could not be compared with the equivalent cases of the AT approach, i.e. those cases where AT predicted high risk, so when obtaining the results, a relevant recalculation is required to compare only with the respective cases of the AT approach. The data set was analysed for data quality and for data integrity and the anomalies were pointed out. Wherever it was possible the data inconsistencies were treated either with an alternative derived logic, or had to be disregarded. To minimize disregarded cases for the calculations the highest risk values were used instead. On one side the data has to be cleansed, but on the other side every effort needs to be made to minimize manual interventions. Either way, the final outcome calculations were definitely affected. To setup the AT approach was straight forward, this involved creating the arguments and the relationships between the arguments called conflicts or attacks, or sometimes called defeats. The conflicts have limitations as they are only used in a binary manner, i.e. either exist or do not exist. To improve this functionality, a future work possibility would be to create a graph which is using a variable scale for the strength of the attacks.

1.6 Document Outline

- Chapter 2 - Literature review - Researches the available specific academic journals and articles, mainly written in the past few years, to get grounding in recent accomplishments in the area of banking, AT and DR. The aim is to list the essential definitions needed in relevant context and determine how other researchers resolved similar problems to this paper's aim.

- Chapter 3 - Design and methodology of the experiment - Establishes the framework and the particular design details which are needed to be able to execute the experiment. The outcome is a process flow with the elements of the AT approach, be executed in a sequential order.

- Chapter 4 - Implementation and results - Executes the experiment following the established design concepts and explains the findings and difficulties around it. The results of the benchmark are determined and the outcome of the experiment is then measured accordingly. The results are displayed with graphs and tables. For highlighting the outcome of the prediction a set of confusion-matrixes are employed.

- Chapter 5 - Evaluation and analysis - Determines whether the experiment results are good results or bad results and compares these with the results of the benchmark. The hypothesis is tested and as the experiment predicts well in 3/4 of the cases, as a final outcome the null hypothesis is accepted.

- Chapter 6 - Conclusion - Summarizes the work which has been carried out, the main findings of the paper, specifically that AT and DR could be used for risk assessment and the corresponding numerical results. Further enhancement areas are also described.

Chapter 2

Literature review and related work

2.1 Overview on mortgages in Ireland

Since the mid 1990s there has been a property bubble in Ireland (Waldron & Redmond, 2014) due to loose regulation in the banking sector and the government's home ownership policies enshrined by the Constitution of Ireland. House prices reached their peak in 2007 and in the preceding recession years house prices fell by 50% on average. This came with high unemployment rates, negative mortgage equities and mortgage arrears. This was a serious problem, which was even pointed out by Bill Clinton the former president of the United States of America on his visit to Ireland in 2011. Davy described various issues around the housing bubble (Davy, 2011). The housing market collapse was the result of the sub-prime mortgage crisis. Davy highlighted the importance of government communicating and informing, as well as the importance of developing and presenting a sustainable economic policy. Information on the housing bubble could have been developed as a warning and presented and implemented in an effective strategy which could have maximized the financial wellbeing of the market. In a housing bubble situation property investors need to refrain from investing more money in housing and need to reduce their portfolio in bank stocks. The stocks are exposed on the stock exchange market when the bubble burst. Since 2007 the global housing and property markets experienced one of the greatest periods of volatility and uncertainty. Initially the crisis was centred on the banking sector and the so-called credit crunch which has it's roots in the mortgage lending practices of the USA which resulted in bank failures and plummeting stock markets. The debt crisis affected Europe soon afterwards, especially Portugal, Greece, Spain, Italy and Ireland. The political landscape was focusing on fiscal discipline through reductions in the level of public services spending, including a retraction of traditional welfare state measures. The global financial crisis has been interpreted as a crisis of neoliberal governance. In Ireland, throughout the so-called Celtic Tiger years, also starting in the 1990s, Ireland's economy was focusing more and more on the property industry and a decade later in the 2000s this was followed with a housing and construction boom, with rapidly increasing house prices, which justified the government's deregulation and a liberal approach aiding developers. The annual construction of houses soared from 19 000 in 1993 to 93 000 in 2006 and the building construction was the dominant sector, accounting for 13.3 % of employment, the highest share among all the OECD countries. After the collapse, the drying up of credit, markets and tax revenue led to a massive banking bailout, which included the establishment of NAMA (National Assets Management Authority) which acquired 88bn EUR of property debt. That would have meant the bankruptcy of Ireland's democratic system, but the EU (European Union) and the IMF (International Monetary Found) bailed out the country with 67bn EUR.

At household level in 2011 Dublin house prices were worth 55% less while rural houses ere worth 47% less. Unemployment rate rose to 14%. Emigration to other countries between May/2009 and Apr/2010 is estimated to have been 65 000 people, followed by another 75 000 people in the subsequent year. In March 2012 there were a total of 760 000 private residential mortgage accounts, of which 77 000 of them were in arrears for more than 90 days, and another 79 000 which were already restructured due to difficulties in repayments. That was a serious amount, considering Ireland's 5 million population, where 2 million people are working and 3 million are not working, even if this figure includes children and elderly (Murphy & Scott, 2013).

Housing prices increased to an alarming trend since 2012. The annual rate of increase[1] was 20% in 2014 and has remained in the double digit since, in 2017 at 1%

[1] URL: https://www.centralbank.ie/docs/default-source/financial-system/macroprudential-

increase per month. Due to job losses in the construction industry during the downturn, there is a serious housing crisis now in Dublin and commuter towns. There were simply not enough houses built since the recession and as the economy is recovering, mainly in the capital, people decided to move to Dublin, or commute from a long distances away, which exhausts the available infrastructure in the capital (roads, public transport), especially in the rush hours. The rental market is heavily affected, too, having only 1500 properties available for rent as of October 2016, which is significantly less than 1% of the total residential units in Dublin. While the Irish banking sector is recovering from the financial crisis, vulnerabilities remain. According to the European Banking Authority this is due to the high ratio of non-performing loans and a high proportion of mortgage arrears. A recent ECB (European Central Bank) report[2] found, that there were 80 000 mortgages for more than 90 days in arrears as of mid 2016, which is a 40% reduction compared to 2013. It is also worth noting, that 50 000 of those cases are in arrears for more than 720 days.

2.2 Predicting defaulting

Credit scoring systems became the norm at banks for assessing credit worthiness of loan applications. The aim of the credit scoring system is to estimate what are the chances that the applicants would run into arrears, or if they can pay back their mortgage, i.e. to estimate the risk of defaulting. The bank's success depends on how accurately can they predict the risk of defaulting. The applicants ability to repay the loan depends on many attributes, but mainly depends on their financial situation (Dukic, Dukic, & Kvesic, 2011).

Fan, Yang, and Zhang (2008) analysed data from a commercial bank in China and their conclusion was, that the bank lends money based on minimizing risk of defaulting and not based on maximising profit.

policy/policy-documents/2016-review-of-residential-mortgage-lending-requirements.pdf?sfvrsn=8

[2] URL: https://www.bankingsupervision.europa.eu/legalframework/publiccons/pdf/npl/stock_taking.en.pdf

2.2.1 Current methods

In traditional banking the assessment of credit worthiness was based on previous experiences (borrowing history) and the internal subjective decision of a business analyst employee. The outcome of this approach was obviously limited. Up two-three decades ago it was simply not possible to use the advantages of ICT (Information and Communications Technology) as the technology devices were not widely available, robust in size and required special skills to operate. As technology evolved, the banks quickly realized the enormous benefits and power of the personal computer devices and the bank's methods to determine risk, which shifted from the traditional subjective approach to a much more technology based objective approach, which now involves calculations and statistics methodologies, which can be used at any stage of the credit management cycle.

One method to calculate the risk of default is the **logistic regression** mode. The way this model works is that it defines the financial indicators of the applicant as random variables which are then simulated. Then the obtained results are used to determine the mean calculation of the risk of defaulting (Dukic et al., 2011).

Another popular method is **neural networks**. Artificial Neural Networks (ANN) are inspired by biological brain activities, where nodes or processing units are connected like neurons which process a signal and pass it on to the next neuron. Those systems learn and progressively improve performance, based on previous examples. This method tends to be useful where the application is difficult to express using rule-based programming (Zha, 2016).

A third widely used method is classification tree analysis, a type of **Machine Learning** (ML), used for classifying data in a structural mapping of binary decisions employing specified algorithms. The analysis started at the trunk of the tree and went on to the branches and then to the leaves. The branches represented attributes and the leaves represented decisions. Sometimes this classification tree is referred to as decision tree, but more precisely it's a type of decision tree which lead to categorical decisions. A regression tree is another form of decision tree, but this one lead to quantitative decisions. L. Rokach and O. Maimon studied the top-down induction of Decision Tree

Classifiers (Rokach & Maimon, 2005). C. L. Devasena researched a practical approach: Adeptness of Memory Based Classifier which was called memory based because it generated hypotheses from training instances, for fast execution (Devasena, 2014).The research predicting bank credit worthiness with ML pointed out, that a robust and automated bank credit risk score with high accuracy is still a major challenge most banks face (Turkson, Baagyere, & Wenya, 2016). As technology evolved a new era of data science unfolded with ongoing research focused on creating and applying several new algorithms with the aim of extracting knowledge from data and finding patterns in data.

Zhu and Wang (2008) described a fourth method the system analysis, where the credit risk assessment is analysed by the data analyst employee based on his or her own knowledge and capability for decision-making, using the **CAMPARI factor analysis**. CAMPARI is an abbreviation and stands for the following: C - Character, A - Ability, M - Margin, P - Purpose, A - Amount, R - Repayment, I - Insurance. The assessment of this method depends on historical data and the employee's subjective judgement and it is more a qualitative analysis than a quantitative analysis.

2.2.2 Business understanding of risk prediction and the Scorecard method

The details in this sub-section have not been presented in literature, but in the medium size Irish bank's internal documents.

To minimise the risk of lending, especially since the economic downturn (Chen, Chen, & Yao, 2010), most banks do a risk analysis, which is based on the collected data from the applicants and their circumstances. The main risk factor of lending is the (re-)payment ability (Yaohua, 2012). The primary influencing factors of risk analysis are the mortgage loan contract, the mortgage property and the financial market conditions (Zheng-xuan & Zu, 2013). Credit risk analysis is playing a key role in stabilizing the bank's investment (Wang, Jeong, Chang, & Ribarsky, 2012).

Beside business interest, there is also a legal requirement to be compliant with the regulations of the Central Bank of Ireland (CBI). Irish banks have to list out the exact steps of their mortgage risk assessment process, how exactly is the credit risk is calculated, and that process has to be review annually, co-ordinated by the CBI.

The definition of loan defaulting by the bank is: if a loan is in arrears for 2 or more months (consecutive or cumulative) and/or contains a specific arrears status and/or is *forborne* (has forbearance flag) and/or has 20% of client exposure for more than 90 days past due.

A mortgage application is flagged as defaulted, at that point of time, if:

- It has an arrears status in (1, 2, 3, 4, 5, 6, 7, 8, 9, C, D, E, F, H, I, J, L, O, P, Q, R, S, U, V, W, Y) (see appendix for code descriptions - Appendix B); or
- It has an arrears multiplier > 2; or
- It is defined as PD $= 10$: unlikely to pay.

A mortgage application might be in arrears only temporarily, but it is still flagged as defaulted, as the flag does not change. However, it is recorded in the core banking system, that the case is not in arrears anymore.

The Scorecard method calculates the Probability of Defaulting (PD) score and at this particular bank there were 12 parameters (risk factors) identified:

- Loan purpose
- Primary occupation
- Secondary occupation
- Employment behaviour
- Single borrower flag
- Self employed flag
- LTV (Loan To Value)
- NDI (Net Disposable Income)

- NDI-ILO (NDI combined with Income LeftOver)
- ICB (Irish Credit Bureau) borrower credit history status
- Region of the asset address
- Amount Drawn To Income

Each of these 12 factors are weighted, based on the WoE (Weight of Evidence). Mathematically, it's a multiplication of the attribute and the weight where the total weight of all attributes add up to 100%.

The final step of the PD-score calculation is an exponential formula, bringing the final score to a scale of 1 to 10, where 1 is the best and 10 is the worst. The formula looks like this:

$$PD = \frac{1}{1 + e^{(4.677 + 2.1175 \times modelscore)}} \times \frac{1.52}{1.50418} \tag{2.1}$$

where the model score is calculated like this:

$$modelscore = \sum_{i=1}^{n} Std.WoE_{var_i} \times weight_{var_i} \tag{2.2}$$

The calculation is done in an analysis software. The run is automated and it is recalculated every month.

The next step of the risk assessment is with the UnderWriting (UW) department. The UW department analyses every case, one by one, in two categories: lending criteria and lending requirement.

(a) Credit Criteria

- The applicant: must be a private person.
- The applicant's age: must be at least 18 years old.
- Loan term: the maximum loan term for first time buyers and movers is 35 years. The loan term is also subject to maximum age at maturity date, as per Credit Requirements (see at (b) below).

- Maximum Loan To Value (LTV): the base of the calculation is the purchase price or valuation of the residential property, whichever is lower. The Max LTV is either 90% or 80%, depending on the location. For First Time Buyers there are other conditions. Total aggregated monetary amount can deviate by 15%.
- NDI% allowed: depending on salary level, and if it's a single borrower or joint borrower, the ND% ranges from 25% to 50%.
- MILO (Minimum Income Leftover): depends on whether the applicants are single borrower or joint borrowers; and on the amount of children; and has to be min. 1500 EUR.
- LTI% (Loan To Income): a limitation of 3.5 times of gross annual income for capping the loan amount for Primary Dwelling Homes (PDH). Total aggregated monetary amount can deviate by 20%.

(b) Credit Requirements

- Purpose: financing the PDH; re-mortgage; further advance (like renovation).
- Buy and hold overall LTV: has to be less than 125% for 2 or more residential mortgages (within and outside the bank).
- Maximum age at maturity date (completion): 68 years old for loan terms up to 30 years; and 65 years old for loan terms longer than 30 years.
- Minimum gross yearly income: For single applicants it's 30k EUR, for joint applicants it's 35k EUR where one of the applicants must earn at least 30k.
- Net monthly income calculation: for payee employees with permanent rights to work in Ireland up to 100% of basic income and maximum of 50% of variable income; for self-employed applicants the calculation is based on the financial accounts and 2 most recent tax returns.
- Affordability assessment: via a stressed mortgage (increased interest rate), for the past circa 6 months overlooking savings, rent payments, previous mortgages.

- Stamp 4 applicant: joint application with an Irish citizen & joint yearly gross income is more than 90k EUR & income split is less than 50% for the Stamp 4 applicant & the Stamp 4 applicant is professionally employed and qualified; every other case the Stamp 4 applicant must be reported as an exception case.

- Net monthly rental income: can be part of the calculation for the approved areas discounted by 30% for taxes and charges.

- Approved locations for 90% LTV: within 50 km of Dublin city centre or Cork city centre; within 25 km of urban areas of Galway and Limerick; within 10 km of Kilkenny, Dundalk, Mullingar, Athlone, Carlow, Clonmel, Greystones, Waterford; other areas are subject to justification.

- Debt consolidation: only for existing PDH customers & for a maximum of 40k EUR & max LTV is 80% & max 2 loans if drawn in the past 3 years & original purpose was works to the property, evidenced with 70% invoices compared to the original loan amount & excludes overdrafts and credit cards.

- ICB checks: must be satisfactory, i.e. no missed financial payments within a period of time; unsatisfactory: A, B, D, F, G, J, K, L, M, P, R, S, T, W (see appendix for code descriptions - Appendix C).

- Maximum number of BTLs (Buy To Let): 2 at any one time (with the bank and outside the bank).

- Scorecards: the PD rating has to be 6 or less.

- Employment length: employees must be permanent and not on probation period; contract employment is allowed for long term or indefinite and likely to have no difficulties finding further work in their sector; self-employed applicants must have at least 2 years history, except professionally qualified applicants who recently became self-employed.

If the applicants qualify for all above, then very likely they will get the mortgage and they will be able to repay, during the term of the mortgage, without any difficulties.

2.2.3 Example of a PD score with the Scorecard Method

The result of the scorecard method calculation is the PD (Probability of Defaulting) which is extracted from the bank's analysis software. A PD result example can be seen at the following table 2.1:

CaseID	**PD**	**ModelScore**	**LGD**	**DefaultingCase**
B164553	3	0.00363622	0.0825	No

Table 2.1: A scorecard example - PD scores

The PD is a scale from 1 to 10, the lower the number the less risky the case is, according to the scorecard model. The PD up to 6 is acceptable, 3 in this particular case is a good result. This is a well-performing case, which did not default in the loan's lifetime, yet, the applicants pay every month their mortgage re-payments. The ModelScore is an interim part of the PD calculation, which is then later transformed to the 1 to 10 scale. LGD stands for Loss Given Default and it means the share of the asset which is lost when the applicant defaults, in other words it is the percentage of the expected loss if the applicant defaults to an irrecoverable level. In this particular case the LGD is 8.25% .

Just for reference, further details about this particular case can be seen in the following tables 2.2 and 2.3 and 2.4. Not exclusively, but all these fields were input parameters of the PD calculation for the scorecard model:

CaseID	**Loan Term**	**Amount Drawn**	**Maturity Date**	**Completion Date**
B164553	30.06	300 000 EUR	01DEC2043	08NOV2013

Table 2.2: A scorecard example - details I

CaseID	Monthly Paym	Primary Occupation	Primary Age	Joint Applicants
B164553	1397 EUR	Quality controller	38	Yes

Table 2.3: A scorecard example - details II

CaseID	Purchase Price	Region Group	Occup Group	Joint Total Income
B164553	375 000 EUR	1	1	86 000 EUR

Table 2.4: A scorecard example - details III

2.3 Challenges in the banking industry

The banks can not fully operate by their free willingness for pure business reasons, because the service they provide affects most families, on a long term. If something goes wrong, that will have a huge effect on the lives of the family members. Financial crimes pose pose another type of risk. The financial market is heavily regulated. If there are restrictions, on the pro side, it should come with something in return. That is, if a bank goes bankrupt, the government can step in to bail out the bank. This ensures sustainability and market stability. (Youqin Pan, 2016) researched sustainability of the financial sector and the finance companies must be client centric, have training & education, have adequate levels of governance, being in control and reduce financial crimes, have established sustainable policies for mortgages and loans. To measure sustainability the financial institutes could comparing their own sustainability levels to the current market trends.

Beside business, market and regulation issues there is another set of problems every bank is facing in Europe and that is related to information, more specifically personal information, as that is sensitive data. A new initiative is called EU General Data Protection Regulation (GDPR)[3] which will be enforced on the 28th May 2018 with

[3]URL: http://www.eugdpr.org/

specific details on how sensitive data should be stored, used and represented. (Tomic, 2010) researched the information based on philosophies from several aspects. The relevant theories and limitations include: the principles of logical reasoning, knowledge base and ethical aspects. T. Taeda pointed out that many traditional problems of argumentation have been already addressed and tackled, such as information needs, emotional spaces, actions, attitudes, intentions, problem situations, cognitive gaps. The following section describes one of these aspects to relate reasoning with mortgages.

2.4 Mortgage defaulting assessment as a form of Reasoning

Zhu and Wang (2008) reviewed the risk assessment with (D-S) Dempster-Shafer's evidence reasoning theory. The reasoning algorithm was developed by Dempster and then further developed by Shafer. D-S run a few different methods and then synthetized the results. The authors prepared a practical case by combining three methods: the system analysis method, the financial ratio synthesis analytic method, and the risk-adjusted return on capital method. The credit assessment was divided into four categories from good to bad: A, B, C, D. The conclusion of each of these methods were expressed with a probability, for example: A - 0.3, B - 0.6, C - 0.1, D - 0, where 0 is the impossible case and 1 is the sure case. First the conclusions of the first two methods (system analysis and financial ratio) were combined in a matrix like a pivot table, using multiplication. The results of the multiplications were then summed up with a corresponding logic. Then this result had to be combined with the third method. The final conclusion was calculated thereafter. Their argument was, that it is safer to use more methods and synthesize its results, as each method had its pros & cons. The disagreements and deviances that way could be eliminated easier between the systems. The general principles of the D-S evidence reasoning theory are similar to the principles of DR and computational AT. The similarity is, that both are using a set of mathematical rules and a categorization for the assessment and both use logic to generate the results, which are the basis for the conclusion. The difference is, that while D-S combines the results of more types of methods, computational AT focuses on generating the results based on the arguments, their relations and their strengths.

Previous researches stated that evaluative arguments were criticized on two main aspects: 1) the researchers tend to focus on very specific aspects of the generation process; 2) empirical tests were not performed (Carenini & Moore, 2006).

2.5 Defeasible Reasoning and Computational Argumentation Theory

2.5.1 Defeasible Reasoning

Defeasible Reasoning (DR) has emerged from Artificial Intelligence (AI) as a sub-topic in the aid of formalising common sense qualitative reasoning.The ability of getting to a conclusion when only partial information is available is an important aspect of mortgage risk assessment, and also in life in general, as information will not always be comprehensively available. In order to support argumentation, humans usually compose a so-called default knowledge, which is the main feature of the reasoning process, even if the premises are not fully known. The pre-conditions, which can not be fully verified are expected to be valid defeasibly only if there is no other contradicting information. If new information becomes available then the conclusion which is derived from the default knowledge has to be re-evaluated and possibly changed (Longo, 2014). This type of reasoning is called Defeasible Reasoning and can be expressed with the following formula:

$$\mathrm{p(x)} : \mathrm{j}_1\mathrm{(x)}\ , \ .\ .\ .\ .\ , \mathrm{j}_n\mathrm{(x)} \text{ - - - - - } > \mathrm{c(x)}$$

where p(x) is the pre-requisite, j(x) is the justification, c(x) is the consequence. If p(x) is known and j(x) is consistent with that knowledge, then c(x) can be subsequenctially concluded. In other words, if the pre-requisite is true and the justification supports that, then it can be assumed, that the truth of the conclusion can be believed. Defeasible reasoning is non-monotonic, unlike standard deductive reasoning, which also means, that adding new premises might lead to removing conclusions and not to adding conclusions (Longo, 2016a). DR is based on the use of rules, which may be defeated by other rules (Antoniou & Bikakis, 2007).

DR constitutes as an efficient rule-based reasoning approach, sometimes with incomplete and inconsistent information. When compared to non-monotonic mainstream approach, then the main advantages of DR are the low computational complexity and enhanced representational options (Kravari, Kontopoulos, & Bassiliades, 2009).

Defeasible Logic is a simple and efficient approach to non-monotonic reasoning which is a particular form of Defeasible Reasoning. The term (i) monotonicity means, that new evidence is not influencing the conclusion, i.e. the set of conclusions monotonically increase whereas (ii) non-monotonicity means the opposite, that in light of new evidence the conclusion changes, i.e. the conclusion decreases in cardinality. An example of non-monotonic inference is *birds generally fly* or *swans are white*. These kind of inferences only hold if there is no other information (argument) which is contradictory. Knowledge is represented in facts, rules, and a *superiority* relation between the rules. Conclusions can be inferred strictly or defeasibly. Example: (i) Facts: predicates *John is a graduate student*; (ii) Strict rules: special predicates *graduate students are students*; (iii) Defeasible rules: special predicates *graduate students usually study very hard*; (iv) Superiority relation: Rule1 is superior to Rule2. In argumentation based reasoning, in any reasoning system, it is important how to represent the conflicting information. The strong opposition is called (i) negation, for example *this pen is blue* negates to *this pen is not blue*. A weaker opposition is called (ii) contrary, for example *this pen is black* can be in opposition with the contrary like *this pen is red* (Panisson & Bordini, 2016).

2.5.2 Applications of Defeasible Reasoning

Panisson and Bordini (2016) refer to the work of Tudor Berariu (Berariu, 2014), who argued, that it is now time to put the theory of Defeasible Reasoning into practice and prove it's usefulness in real applications.

L. Longo used an extensive framework which was built on defeasible reasoning and implemented with argumentation theory (AT) for modeling human mental workload (MWL) across different areas of human-computer interactions (HCI) (Longo, 2015). Particular applications of MWL were studied for medicine (Longo, 2016b).

L. Longo and P. Dondio prepared a summary paper on defeasible reasoning and argument-based systems which described medical field scenarios with the aim of framework development for inference, dialog, learning and decision making for these medical areas: group decision support, decision on the best treatment for patients; genetic counselling; predicting the recurrence in breast cancer patients who had surgery; investigating treatment efficiency; organ transplant (Longo & Dondio, 2014).

AT gained importance in healthcare due to its modular and intuitive way of gathering and aggregating evidence and making rational decisions. It was shown how to represent available clinical evidence with the AT framework and how to compute the argument's justification statuses with semantics (Longo, Kane, & Hederman, 2012). The role of AT and DR were investigated for decision making support in the healthcare sector on breast cancer recurrence prediction, compared to Machine Learning (ML) classifiers. AT is not a learning based neither probability based paradigm, but knowledge based paradigm and deals well with new evidence and contradiction. ML requires large data sets and long computational times to learn from previous examples (Longo & Hederman, 2013).

E. Bellucci in her study integrated Artificial Intelligence, Argumentation and Game Theory to develop an online negotiation and dispute resolution environment, for Alternative Dispute Resolution (ADR), which was for resolving disputes with other means than litigation (legal action), such as arbitration or mediation. ADR pushed the dispute resolution away from the court room and Online Dispute Resolution (ODR) follows this trend (Bellucci, Lodder, & Zeleznikow, 2004).

S. Brasil used Artificial Intelligence (AI) rules and principles in Law Argumentation Theory and built a model, which allowed reasoning with conflicting legal principles. His study aimed for treating vagueness and collision of principles in the aid of establishing rationality in legal argumentation systems (Brasil Jr, 2001).

A. Bikakis applied Contextual Defeasible Logic (Bikakis & Antoniou, 2011) and Defeasible Contextual Reasoning (Bikakis & Antoniou, 2010) to Ambient Intelligence. Ambient intelligence is a user-centric paradigm, it aids the user pervasively, non-intrusively and transparently, using several methods from Artificial Intelligence. The research of A. Rakib supported contextual non-monotonic reasoning (Rakib & Haque, 2015).

M. Cristani et al. applied Defeasible Reasoning (DR) to an electric consumption model, using defeasible rules, conflicting rules and exception rules, aimed for energy savings in a system of devices (Cristani, Tomazzoli, Karafili, & Olivieri, 2016).

N. K. Janjua is described a web-based Intelligent Decision Support System (IDSS) and used defeasible reasoning based argumentation and supported it's usefulness when some underlying information is missing (Janjua & Hussain, 2011).

C. I. Chesnevar and A. J. Maguitman in their paper presented a defeasible argumentation based solution for resolving web search queries. The success of search engines like Google not only depend on a large number of indexed pages, but also on the quality of the search results returned (Chesnevar & Maguitman, 2004). T. T. Quan et al. performed a similar research on online digital libraries (Quan, Luong, Nguyen, & Cheung, 2015).

2.5.3 Computational Argumentation Theory

Defeasible Reasoning (DR) (Eemeren, Grootendorst, Johnson, Plantin, & Willard, 2013) and Defeasible Logic (DL) (Nute, Henderson, & Hunter, 1997) have strong relationship with Argumentation Theory (AT) as AT provides state-of-the-art computational models and techniques to implement DR. Therefore AT has gained interest in Artificial Intelligence (AI), inspired by human reasoning. AT is a multi-disciplinary research subject, which derives from various other disciplines like sociology, psychology, philosophy, law and linguistics. The focus of AT is how pieces of evidence interpreted as arguments can be represented, supported and discarded in a defeasible reasoning process (Longo & Dondio, 2014).

Argumentation is the process of constructing the assumptions for solving the analysed problem, which involves the detection of conflicts and finding a solution. There are arguments, which are *for* the assumption and there are arguments which are *against* (Vagin & Morosin, 2013).

The classification of AT (Longo, 2016a) can be seen at the following table 2.5.

	Monological	**Dialogical**	**Rhetorical**
Structure	Micro	Macro	Persuasive
Foundation	Arguments as tentative proofs	Defeasible reasoning	Audience's perception of arguments
Linkage	Connecting a set of premises to a claim at the level of argument	Connecting a set of arguments in a dialogical structure	Connecting arguments in a persuasive way

Table 2.5: The classification of argumentation models

Monological models are concerned about the internal structure of the arguments and their components, i.e. premises, rules, conclusions and their relations, the construction of arguments.

Dialogical models highlight the argument conflicts and their resolutions in the arguing process, conflict management.

Rhetorical models focus mainly on the argument's perception by the audience and the means of persuasion how the audience is convinced.

When an argument system formed it's internal construction of arguments the non-monotonic reasoning way (Longo, 2012) then that is considered as a micro structure. If on top of this the supportiveness for or against a conclusion is added, too, then that is considered to be a macro structure.

Argumentation systems are generally built on five layers:

1. definition of the internal structure of arguments

2. definition of conflicts between arguments

3. evaluation of conflicts and definition of valid attacks

4. definition of the dialectical status of arguments

5. accrual of acceptable arguments

These five layers are described in the following sections.

Layer 1: Definition of the internal structure of arguments

The representation of the internal structure of arguments are usually represented by monological models and the following formula can be used:

Argument: P_1 , P_2 , . . . , P_n - - - - - > C

where P's are the premises and C is the conclusion.

L. Longo described the classical model of Stephen Toulmin by highlighting the model's key elements of an argumentation and knowledge representation. Toulmin (Zeleznikow & Stranieri, 1998) described six components as per following:

- Claim (C): conclusion, might be controversial
- Data (D): statements of facts and beliefs
- Warrant (W): statement that justifies the Data
- Backing (B): information that backs up the Warrant, when the Warrant is challenged
- Qualifier (Q): statement of the degree of certainty
- Rebuttal (R): a statement where the conclusion might be defeated

A practical example of Toulmin's model can be seen on the following figure 2.1:

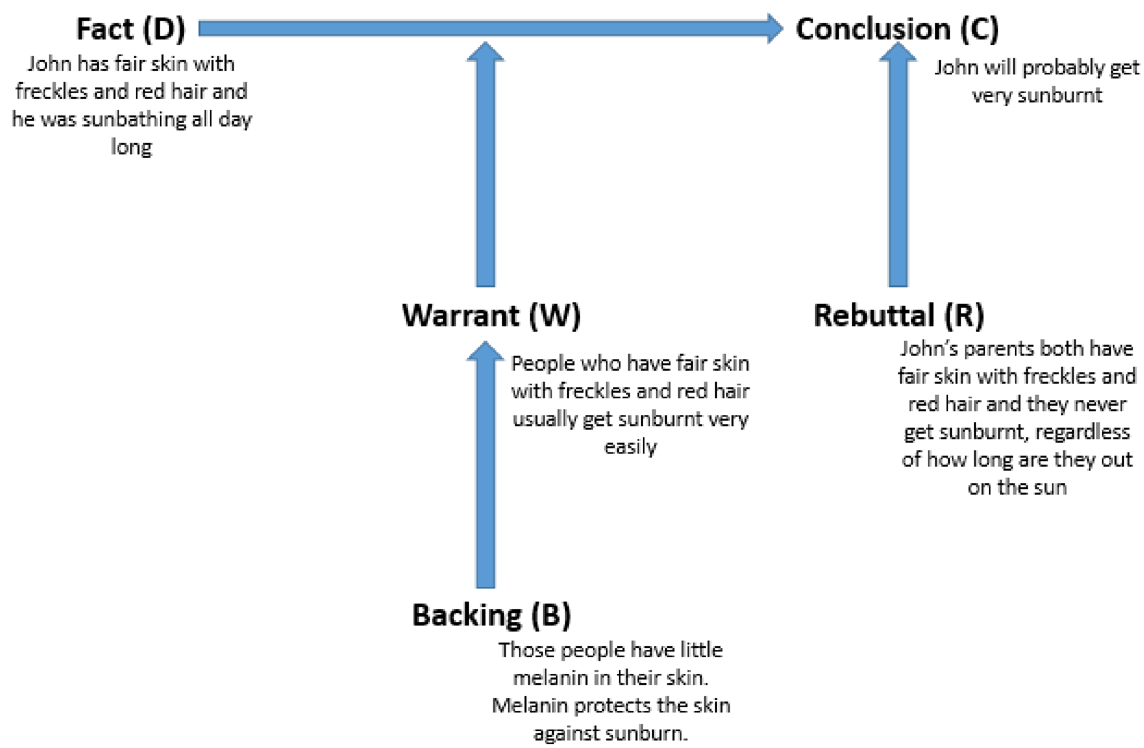

Figure 2.1: Example of Toulmin's Argumentation Flow Chart

Layer 2: definition of conflicts between arguments

The relationship between two arguments is called a conflict, or attack, or defeat (Bisquert, Cayrol, Saint-Cyr, & Lagasquie, 2013). Those arguments which appear to be valid, but are invalid are called fallacious arguments. The conflicts can be categorized as per the following three classes (Rizzo, Dondio, Delany, & Longo, 2016; Rizzo & Longo, 2017):

- Undermining attack: occurs when the premise of an argument is attacked by an argument which in conclusion negates the premise
- Rebutting attack: happens when an argument negates the conclusion of another argument
- Undercutting attack: is performed when an argument uses a defeasible inference rule and is attacked by another argument which says that there is a special case which does not allow the execution of the rule itself

Layer 3: evaluation of conflicts and definition of valid attacks

Conflicts between arguments are important notions, but they do not always determine the success of the attack. Generally an attack has a binary form of relationship between two arguments, but it is also possible that an attack has a weaker form called defeat, or a stronger form called strict defeat. This strength can depend on various things, for example the expertise of the observer; or the reliability of the tests performed. Evaluating attacks can happen via the following two ways:

- The strength of arguments (or preferentiality of arguments)
- The strength of attack relations (or preferentiality of attacks)

Strength of arguments: is establishing whether an argument can be successfully defeated which is based on the inequality of the strength of arguments between arguments and counter arguments. This has to be considered as part of the computation for the sets of arguments. For example if argument X undercuts argument Y then X is a successful attack if Y is not stronger. If the strength of an argument can be expressed with a numerical value (for example X = 0.6 ; Y = 0.4), then an attack from X to Y is only successful if the strength value of X is ranked higher or equal to the strength value of Y. Strength of attacks: is about attaching weights to the to the attack relations, instead of the arguments. The way of doing so is by introducing the notion of inconsistency budget, which quantifies the tolerance level of the inconsistency. Introducing an inconsistency level for X, the attacks can be disregarded up to the total weight of X. By increasing this threshold, more attacks would be disregarded. As a consequence, a preference order is formed, where a lower inconsistency budget is preferred (Longo, 2016a).

Layer 4: Definition of the dialectical status of arguments

Layer 3 focused on the strength of the arguments, but in that phase it is not known yet which arguments can be seen as justifiable. The final state of each argument is highly dependent on the interaction with the other relevant arguments and therefore a dialectical status outcome is needed. Layer 4 aims to determine this outcome. The way to achieve this outcome by splitting the arguments into two sub-groups, namely into those arguments which support an action an decision and those who do not. On some occasions a third sub-group might be used as well for those arguments, which get to a so-called status: undecided. This is not a strict classification, the number of groups could be increased.

Modern implementations of calculating the dialectical status of arguments are based on Phan Minh Dung's abstract Argumentation Framework (AF), which allows for comparison between different systems by defining a set of abstract arguments where the internal structure of the argument is emitted and a set of defeat relations decide which arguments get accepted and which get rejected (Dung, 1995b). Dung et al. developed a general theory for argumentation and showed that logic programming and nonmonotonic reasoning are different forms of argumentation (Dung, 1995a). In Dung's AF environment the arguments are represented with nodes and the attacks are represented with arrows. For example:

A < - - - - - B < - - - - - C

Argument B defeats argument A if B is a valid reason against A. However, only by looking at the defeater of an argument would not be enough to be able determine the final outcome of the argumentation, because the defeaters could have other defeaters, which also need to be looked at. Looking at the example above B is attacked by C and as C is not attacked by anything therefore C is accepted and because C attacks B, B is rejected. As B is not a valid argument anymore for A, so A is finally also accepted. In other words C reinstates A (Longo & Dondio, 2014).

In an AF to be able to decide which arguments are accepted and which are rejected a formal criteria is necessary. This is called acceptability semantics, which finally specifies zero or more so-called extension, which are the set of acceptable arguments. The labeling approach can take three different values (Dauphin & Schulz, 2014):

- IN - for accepted arguments

- OUT - for rejected arguments
- UNDEC - for undecided arguments, which can be neither accepted, nor declined

To assign these values to the arguments they have to fulfil the following two conditions:

1. an argument is labelled IN, if all it's defeaters are labelled OUT
2. an argument is labelled OUT, if it has at least one defeater labelled IN

Furthermore, the definition of a conflict-free argumentation is: if and only if it does not contain any argument A and B where A defeats B. A set of arguments Args is so-called defend argument C if and only if each defeater of C is defeated by an argument in Args. This concept is called Complete Semantics, and aims on computing complete extensions (acceptable arguments). There are other semantics. In Complete Semantics more than one complete extension can exist, but in a more sceptical approach where exactly only one extension exists is called a Grounded Semantics (Bisquert et al., 2013). At Grounded Semantics and Complete Semantics the IN labelled arguments are minimised, so are the OUT labelled arguments and the UNDEC is maximised and also can be empty. If instead of this sceptical approach a credulous approach would be desired that is known as the Preferred Semantics. This approach instead of maximising the UNDEC arguments it maximises the IN arguments and OUT arguments, based on the notion of so-called admissibility. An argument set is admissible if and only if it is conflict-free and defends at least itself. The empty set is always admissible as it's conflict free and has no defeaters. It always exists at least one preferred extension (accepted argument). Every grounded and preferred extension is a complete extension. On the example above the admissible sets are {C} and {A,C} whereas {B} and {A} are not admissible as they are attacked. Only one preferred extension exists: {A,C}. L. Longo mentioned in his article that P Baroni et al. describe several other types of semantics (Longo, 2016a).

Layer 5: accrual of acceptable arguments

Multiple acceptable extensions of arguments can be calculated from the previous layer 4 dialectical status of arguments, but sometimes it is desirable to have a single result which is the aim of layer 5, to accrue the acceptable arguments. Different strategies for computing can be developed, for instance based on credibility of extensions (acceptable arguments), based on the strength of arguments, or based on a preference list, etc.. At this stage any logic and algorithm could be applicable, but that should be selected, whichever serves the decision making the best (Longo, 2016a).

The determination of the accepted arguments under a specific semantics can be time-consuming and tedious (happening in several steps) (Bilo Doumbouya, Kamsu-Foguem, & Kenfack, 2016).

2.6 Summary of literature review, gaps and issues

2.6.1 Summary

Ireland has been in a propery bubble since the Celtic Tiger years, since the mid 1990s where property prices continuously increased and house prices went up, mortgages were available with very little restrictions and zero deposit. In 2008 recession hit the country which ended this era and property prices fell to half in the following 2 years. Banks had to be bailed out to stabilize the economy and regulations had to be implemented for bot the government and the banks. Risk analysis became a key component of mortgage lending. Several techniques are tried Machine Learning, Artificial Neural Networks, but the norm is the Scorecard method, which is calculated with a formula, based on credit criteria and credit requirements as input variables. The output is the prediction of Probability of Defaulting, a numerical value representing a risk level which ranges from low to high. Risk assessment was reviewed as a form of reasoning using the Dempster-Shafer evidence reasoning theory, which used more methods and then synthesized them. Defeasible Reasoning and computational Argumentation theory was introduced and described in details with the 5 typical layers and practical applications in healthcare, mental workload, law, decision support, search queries and electronic consumption.

2.6.2 Gaps and issues

The goal of literature review was to find out how is it possible to improve the PD prediction. It was reviewed if there are any researches available on using AT and DR for predicting the risk of lending and there is a problem in the literature as there is not a lot of work done so similar cases were studied and there are inconsistencies in pieces of evidence or the research method can not be directly applied. The limitations of the scorecard method are, that it is complex to calculate, it is using a static formula, it is data driven, the categorization of it's input parameters might be difficult to interpret and may not be based on facts and logic, deviation and exception cases might break the rules, further processing of the lending may be difficult to be incorporated.

2.6.3 Research question

To address the gaps and to improve PD prediction the following reserach question is formed:

Q: Can the use of Defeasible Reasoning and Computational Argumentation Theory increase representation and assessment the risk of mortgages, compared to the scorecard method?

In order to answer this question an experimental study employing DR and AT is designed in the next chapter, which shows the circumstances and setup details of DR and computational AT how could those be used as another method to compute the risk of mortgage loan defaulting.

Chapter 3

Design and methodology

3.1 Design overview

The design chapter is about setting the scene for the experiment, describing it's environment and planning how the experiment will be performed. This chapter is focusing on the preparation work what it takes to get ready for the experiment and describe how it could be executed in a controlled manner. Initially, a testable hypothesis is set, which is then further described by six thesis objectives as guidelines for preparing and executing the experiment and this chapter is about expanding objective 2: designing a solution. The design is split into seven design phases, seven layers of the experiment, which need to be executed sequentially in this order: 1) Data understanding, 2) Data preparation, 3) Internal structure of arguments, 4) Conflicts of arguments, 5) Dialectical status of arguments, 6) Accrual of the acceptable arguments, and 7) Comparison. The 7 layers start with the analysis and preparation of the data set, which includes a data quality report on a representative sample extract for the most important features. Then the paper describes the details how to model mortgage assessment with Argumentation Theory as part of the last 5 layers. Then it is described how the results can be gathered and what are the essential specifics to be able to compare the bank's regular method (scorecard) with computational AT, i.e. the aspects of how to compare the computational AT results with the benchmark and what are the criteria's to ensure that the same specifics are compared. The last layer makes a recommendation for selecting the relevant semantics which can be applied for this type of experiment from a list of available semantics.

3.2 Hypothesis

The calculation of the risk of mortgages can be seen as a defeasible reasoning activity: in this sense it is subject to defeasible arguments, which is resolved with DR (Defeasible Reasoning). These activities can be modeled in practice using computational AT (Argumentation Theory).

The hypothesis set in this study is:

- H0 (Null hypothesis): The inferences produced by a model built using computational AT (Argumentation Theory) are expected to over perform the Scorecard method, in terms of default loan prediction.

As a consequence the alternative hypothesis is:

- H1 (Alternative hypothesis): The inferences produced by a model built using computational AT (Argumentation Theory) are equal to or less predictive than the Scorecard method.

The prediction is the variable what we need to calculate.

3.3 Design phases

As an expansion of the thesis objective 2 (designing a solution) a wider set of tasks are displayed in 7 phases, 7 layers of the experiment, which can be viewed on the following flow chart diagram 3.1 :

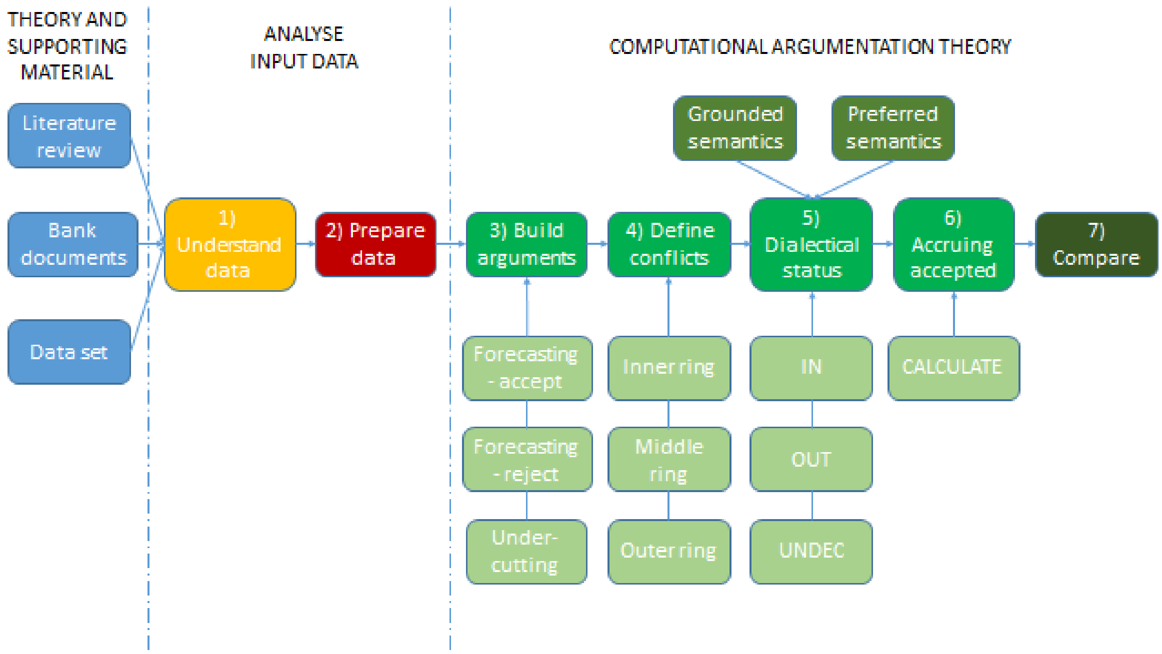

Figure 3.1: Design overview

The design phases are a list of actions, to design a solution in order to resolve the hypothesis, which need to be executed in this sequential order:

- Phase 1: Understand available data
- Phase 2: Prepare the data (dataset size, cleansing data)
- Phase 3: Define the internal structure of arguments
- Phase 4: Define the conflicts of arguments
- Phase 5: Define the dialectical status of arguments
- Phase 6: Define the method for accruing the acceptable arguments and produce a final inference
- Phase 7: Compare the final inference AT approach versus Scorecard method

3.4 Design details of the research phases

3.4.1 P1: Data understanding

The dataset analysis should be starting with a quick overview to verify that the data is readable, it is not corrupted and it makes sense, for example check the ranges of dates for contract date, to see if there are any outliers. From the available columns it would be advisable to prepare a shortlist to determine which columns would be needed for both the model and for verifying the results. In that sense the definition of defaulting has to be established, if it covers only irrecoverable cases, or also those which were defaulting temporarily. Irrecoverable means that the application permanently defaults, so payback of the mortgage is not possible and usually this leads to a legal action. Defaulting, in general, means a temporary status, and in the better case it means, that during the lifetime of the mortgage there were 2 or more consecutive payments missed, which results in being in arrears for over 90 days. Performing means, that the mortgage is on track, payback payments from the applicants are continuously received, usually on a monthly basis. In other words what needs to be established is what is a good result and what is a bad result, in the data.

There should be no sensitive data in the data set, but it would be worth double checking and if there is, then report and the sensitive piece is to be destroyed immediately.

Data Quality report should be prepared using a sample data, for example a 500 records random extract which represents the whole set.

3.4.2 P2: Data preparation: Filtering and pre-processing

Cross industry standard process is data mining.

Once the data set is available, in a readable and understandable format, then there is usually further work required, before the data set can be used. First the platform needs to be decided what software will be used and verified if the selected platform is suitable for the task. A database software would be more advanced than a simple worksheet application and therefore preferred. On the dataset it is highly recommended to start with a check for duplicated records. If the database option is chosen, then the duplication check can be easily performed in SQL server using the SELECT command and compare the amount of returned rows with SELECT DISTINCT. To have a clean data set, it is also advisable to replace empty strings with NULL values. This is more a database classification than a filtering activity and even though this step is certainly not essential, but it helps to make it easier to perform some operations and also a wider variety of functions are available in SQL Server. If there are some values missing those values need to be treated, either to find an alternative value, or define a logic to overcome missing values, or as last resort disregard that record, but this should be avoided as much as possible. Either with a derived calculation, or with an equivalent alternative feature, if available. If it is not available, then it is important to make sure to use a logic as error handling for the further calculations, i.e. generating the arguments. The worst case scenario is, that those records have to be disregarded for further calculations, but this should be avoided as much as possible, to achieve a high level of completeness.

In summary, the data preparation phase should cover all activities to construct the final data set.

After all above mentioned adjustments to the data set have been done the data should be ready to be used for defining the arguments for DR.

3.4.3 P3: Internal structure of arguments definition

The arguments definition is about building the arguments. To build the arguments a knowledge base is to be created, which acts as a rule engine. The output of the argument calculations could be categorised as per following:

- Dependent variables - Risk Categories
- Independent variables - Risk Level values

The dependent variables (risk categories) could be represented numerically as independent variables (risk level values), as per following table 3.1 :

Risk category	**Risk level**
Low Risk	1
Medium-Low Risk	2
Medium Risk	3
Medium-High Risk	4
High Risk	5

Table 3.1: The Risk Categories

The preparation phase of creating the arguments initially involves the creation of a set of arguments. One argument can be represented by one node. For easy identification the nodes should have a naming convention. For example the name could start with the letter R (for aRgument) + then the argument's listing number + and then the risk level indicator code. An exact example can be seen on the following figure 3.2 :

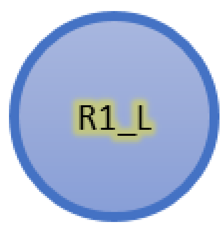

Figure 3.2: Argument visualisation

Each of the arguments could be represented with a circle (node) to visualize the argument and the argumentation's rule can be either embedded or calculated separately and only displayed with a name. As it is possible to have some complex embedded arguments, the arguments could be pre-processed using SQL language before they are fed into the computational Argumentation Theory model.

Each of the arguments should represent an area of the risk assessment, used as an input parameter or input factor, however, also a combination of input factors can be used. These factors should be then transformed to a computation, which can be the base to calculate the argument's output. Wherever it is possible, the outcome of the argument itself could be categorised on a scale. Adequate level of the scale is advised. The following representation is an example what could be used. A scale of [1..5] for Risk Levels (RL), where 1 - Low risk; 2 - Medium-low risk; 3 - Medium risk; 4 - Medium-high risk; 5 - High risk.

Once the set of arguments are defined the concept of the argument interaction needs to be created and to achieve that the set of arguments could be categorized. For risk assessments the whole set of arguments could be organized within 3 logical rings, as per the following figure 3.3 :

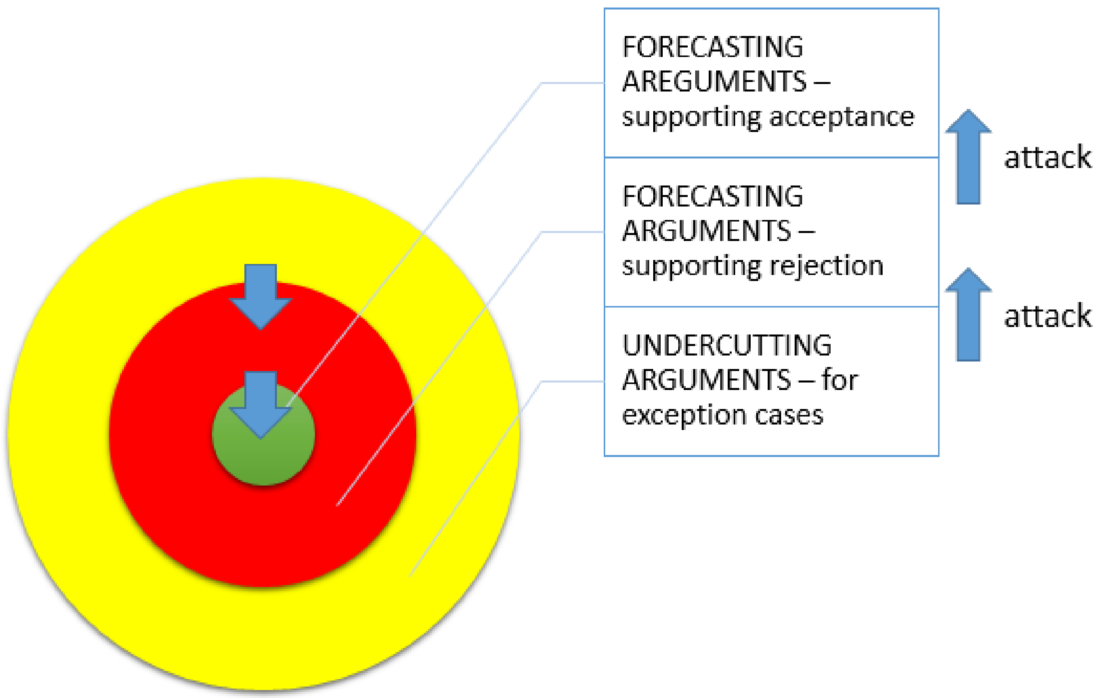

Figure 3.3: Arguments definitions - conceptual model - on 3 rings

The innermost ring should be the initial forecasting set of arguments which are supporting the mortgage application for acceptance, meaning that the expected results are between Low Risk and Medium-High Risk. The middle ring should be used as a set of forecasting arguments which are supporting the rejection of the mortgage application, so those cases where the expected result is High Risk. The outermost ring should have a set of arguments, which are called undercutting arguments, to handle the exception cases. If the data qualifies for the execution of the undercutting arguments, then the risk levels are lowered from the rejection levels to the acceptable levels and as a consequence, the final result and the conclusion of those set of arguments could change.

The setup design details of these three categories of arguments are described in the following sub-sections.

Forecasting arguments - supporting acceptance

The model representation for the initial set of forecasting arguments for computational AT which support acceptance is as per following. An individual argument is created for each of the set of arguments for each of the available acceptance risk levels, for example: Low Risk; Medium-Low Risk; Medium Risk; Medium-High Risk. If all these arguments are accepted, then the applicants should get the mortgage approved and chances are high that the applicants will be able to pay back their mortgage on time.

Forecasting arguments - supporting rejection

In this phase, the risk category High Risk is in opposition with all other lower risk levels, as this is that category, where the mortgage is refused, because the applicants probably will not be able to pay back the mortgage. It can not happen that a forecasting argument is supporting acceptance and rejection in the same time.

Undercutting arguments

To continue with the argumentation, in this model, there should be exception arguments for those risk level High Risk cases, which are in contradiction with the sets of forecasting arguments supporting rejection. There should be set of arguments created which are undercutting, i.e. lowering the High Risk arguments from rejection levels to acceptance levels, otherwise without these sets there is no real argumentation going on. Without these set of arguments all the possible points of views in argumentation (Expert System, Grounded Semantics, Eager Semantics, Ideal Semantics, Preferred Semantics, Stable Semantics, Semi-stable Semantics, Admissible Semantics) would have returned the same result values. These exception conditions are also happening in real life. For example, the UnderWriting department at the bank would still consider some applicantions for the mortgage to be awarded, even though it would not fully qualify, but if it is a convincing and strong case, the applicants have a secure job, the property to mortgage is on a great location, etc. then this case can still be awarded with the mortgage loand and handled as an exception.

Once the sets of arguments are defined then their relationships have to be established, as per next section.

3.4.4 P4: Conflicts of arguments definition

The definition of conflicts is about the setup and connections between the arguments, namely: which argument is attacking which other arguments. The conflict is also called defeat or attack. The attack can be represented with an arrow, which points from one argument (node) to another argument, as per the following figure 3.4 :

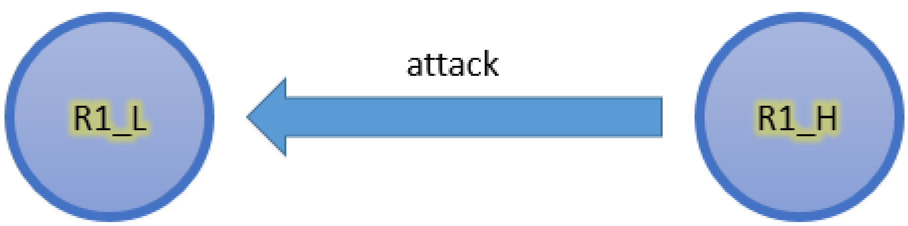

Figure 3.4: Attack visualisation

In this case R1_H argument is attacking the R1_L argument. The attack itself is represented with the arrow. The arrow points from one argument to another. That argument where the arrow starts (R1_H) that is the argument which is attacking, and that argument, where the arrow finishes (R1_L), where the arrow points to, is the argument which is attacked. It is possible that one argument is attacking more arguments, and it is also possible, that one argument is attacked by more arguments.

For this model the definition of conflicts, namely the direction of the conflicts should be setup according to the 3 rings conceptual model as described earlier at the arguments definition section 3.3 and then the model could be represented like the following figure 3.5 :

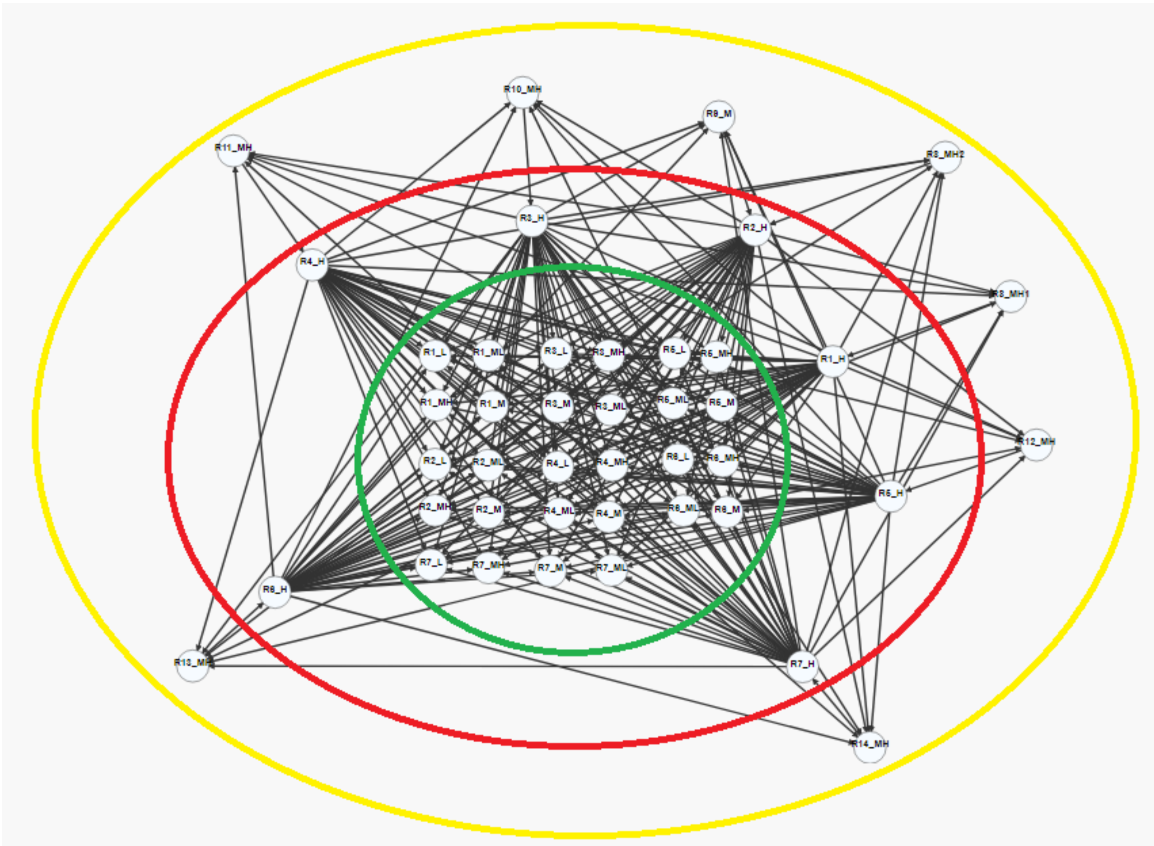

Figure 3.5: Conceptual model representation - attacks

If we consider to have 7 sets of arguments for forecasting acceptance, 4 risk levels each; 7 forecasting arguments with 1 risk level each; and 7 undercutting arguments, then the argument relationship, the attack details can be described on the conceptual models in more details like the following:

i) Innermost ring (green): could contain the 7 set of forecasting arguments (R) for acceptance, which are multiplied by the risk levels (RL) between [1..4] that would result in 28 arguments: (7 R * RL[1..4] = 28 arguments). In the proposed model these arguments do not attack anything, initially these arguments are the expected to have positive outcome cases, meaning that the risk of defaulting is low, in other words no mortgage re-payment difficulties.

ii) Middle ring (red): could have the 7 sets of forecasting arguments for rejection, which represent high risk [5] cases and these would result in the setup of 7 forecasting arguments for rejection: (7 R * RL[5] = 7 arguments). These arguments should attack all the 7 set of forecasting arguments for acceptance.

iii) Outermost ring (yellow): could contain the additional 7 set of undercutting exception arguments (ER) resulting in 7 arguments: (7 ER * RL[5] -> RL[4] = 7 arguments). Each of these arguments would attack the relevant forecasting arguments for rejection (which are high risk) and also should be attacked by every other forecasting arguments for rejection. If the case qualifies to be applied and if it is successful, then the undercutting arguments will be overruling the relevant forecasting arguments for rejections, on a way described in the following sections.

For the initial setup of the model the strength of the attacks should not be considered. For this study binary attacks are recommended to start with, so instead of a scale for strength, every attack should be either full strength or virtually zero strength, if there is no attack.

3.4.5 P5: Definition of the dialectical status of the arguments (dung)

In the dialectical status of the arguments phase in the design, it should be decided which arguments could be accepted and which arguments could not be accepted. At this stage, the logic should be the same for all, but it should be executed on case level, i.e. individual mortgage application level.

The labeling logic of the arguments could looks like this:

i) The labeling starts at the outside perimeter: to find all the initial arguments, which are not attacked and label them IN.

ii) If there are no initials present, then label all of them as UNDEC and terminate

iii) Run a cycle:

Repeat

- label as OUT all those nodes which are attacked by a node which is labelled as IN

- label IN all those nodes which are attacked only by OUT nodes

- continue the loop if there are still nodes which are labelled as IN in the previous steps

End of the loop

iv) Label all the unlabelled arguments (if any) as UNDEC

Practically, as a visual representation of the outcome of this phase, we are going to end up with 3 different results for each of the arguments, which are: IN; OUT; UNDEC. For each case in the dataset, these arguments could be coloured which represent the result. Those arguments which are accepted (IN) could be highlighted with green colour, those arguments which are rejected (OUT) could be highlighted with red colour and those arguments, which are neither or do not apply, which are undecided (UNDEC), could be highlighted with white colour. An example of the visual representation of the dialectical status of arguments can be seen on the following figure 3.6 :

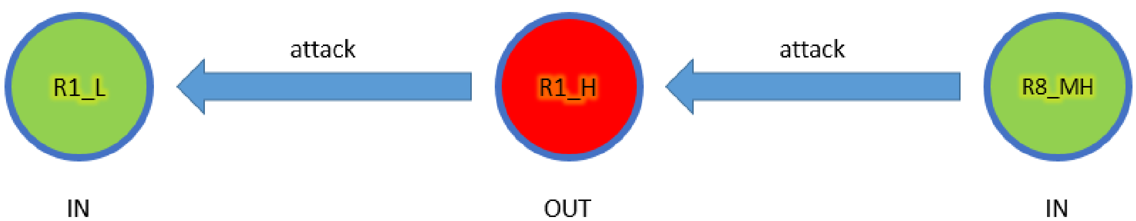

Figure 3.6: Visual representation of the labeling - DUNG

3.4.6 P6: Accrual of acceptable arguments

The next step in the process is to aggregate the outcome of all the relevant arguments, based on which arguments are accepted and produce the outcome as the result for a mortgage application: PD in this study.

The accrual should be an ifelse: if there is an argument with High Risk then PD = 5, else PD = average of the arguments not with high risk. For each case in the dataset the Average (arithmetic mean) of the accepted arguments could be calculated, which should give the final PD score for the paper.

There should be one more last step in the process. The PD score, if it is defined with a scale, for example from 1 to 5, that scale should be transformed to a binary outcome: either yes, the mortgage is accepted and we can say that the applicant is capable of paying back the mortgage; or no, the mortgage is not accepted and the applicant possibly will not be able to pay back the mortgage. It is possible to define more different mappings for this exercise, a particular example for two different ways of aggregation:

Version 1 - less strict: Risk levels 1 to 4 is YES, Risk level 5 is NO.

Version 2 - more strict: Risk levels 1 to 3 is YES, Risk levels 4 to 5 is NO.

3.4.7 P7: Comparison

The goal of the comparison is to determine whether the bank's PD calculation, the Scorecard method is predicting better, or the accumulation of the defeasible reasoning is predicting better. The dataset should have a flag, which is marked, if a mortgage application case was in arrears or not. However, by definition, that flag could be applied also for those cases, which were in arrears, but are not in arrears anymore. Special attention is advised when handling this data, as the defaulting status purely could be only a temporary status. From the business point of view, the real problem is, if there is a case which is in arrears on a longer term (more than 90 days), especially if that case will lead to be totally irrecoverable. The Central Bank of Ireland is also tracking these applicants and they will get marked there as well if they miss two or more payments within a period of time. Being in arrears will influence the applicant's ability to acquire any further loans in Ireland. The comparative study should performed on the following way: the aggregated PD outcome of the computational AT should be compared with the outcome of the bank's PD outcome by the bank's scorecard method, which should lead to the final set of conclusions and eventually to the hypothesis testing. If there is no argument that should be acceptable or if the calculations will not return any result, that could happen and then those interim results have to be handled with a logic for the final result calculation. These events could happen in case no argument is capable of defending itself from all the other attackers. The logic to treat these cases for the final calculation could be included on two ways:

i) consider those cases as high risk for the risk of defaulting: NO, expected not perform well (bad cases)

ii) consider those cases as low risk for the risk of defaulting: YES, expected to perform well (good cases)

Considering a set of arguments where there are two possibilities, either to accept arguments (A, C) or to accept (B, D). If (A, C) is chosen, then they are capable of defending themselves from (B, D) and vice versa. In this case the final score has to be derived with a calculation, and it could be calculated as the score of the set with the highest number of arguments, or the average of all sets if they have the same size.

3.5 Strengths and approach taken

The approach recommends to produce a Minimum Viable Product (MVP) first. This paper intends to investigate what is possible to describe finance areas, namely mortgage prediction with Argumentation Theory and Defeasible Reasoning, using familiar terms from human reasoning. The MVP consists of a workable and visually represented model, which produces useable and interpretable output and clearly identifies each mortgage application case whether it would default or whether it would perform well. The way to achieve this is via analysing the existing business process, namely the scorecard method and see the exact details how is it calculated, at all three stages: the credit requirements, the credit criteria and at underwriting. This can be used as the benchmark. To determine whether it is possible to use DR and computational AT for loan risk prediction, the available literature review needs to be translated to actionable items, which is the 7 layers as described in the design phases of the paper. The data set seems to be sufficient enough to be used for further calculations, containing 60k+ applications, which is still a small number compared to the big leading banks, but it is still an adequate size to compose the reasoning details for the computational AT approach. The content of the data was reviewed with data quality reports and even though there are some missing or inconsistent values, probably due to original data entry issues to the core banking system, but the majority of these gaps can be treated with a logic, a derived calculation, however, that is only a workaround and these workarounds will still be affecting the results, the final outcome of the analysis, however, it is believed, that not significantly.

Chapter 4

Implementation and results

This chapter describes how the designed solution in the previous chapter was implemented as an expansion of the 3rd thesis objective (execution of the experiment). There were 7 design phases defined, each layer is implemented to the Argumentation Theory (AT) framework and their results are obtained, starting with the benchmark, the bank's method.

4.1 Layer 1. - Data understanding

The dataset which is in use in this research study has been gathered from a medium size Irish bank on mortgage applications. The data contains the most important metrics about the mortgage applications, like asset location, dates, amounts, loan terms and the calculated PD score, etc. Two data sets were received, both were produced by the bank using the same pre-defined query by the ICT (Information and Communications Technology) Department. The first dataset has 6535 records, the second dataset has 66 196 records. The first data set contains mortgage applications between 2012 to 2015. This second data set contains cases as far back as April 1989 and until January 2017. There is a huge overlay between the 2 data sets, namely 6382 cases of the first data set are part of the second data set, in other words 97.7% of the first data set is part of the second data set. Because of the high percentage of the overlay there are two options, either disregard the first data set completely, or merge the 2 data sets.

By looking at the date ranges, that tells us that the date range is a full overlay, and it is not possible to know why is there a 2.3% difference. There could be a mistake in the first query and because the second data set was received later, the first option would be recommended, so disregard the first data set and use only the second set. Looking at the structure of the data sets, both datasets have a total of 342 columns, each, exactly the same columns. To execute the experiment only 14 columns of these 342 columns would be needed, mainly for the computational AT calculations of this study, but an additional 5 columns would be also needed to be able to verify the results. In summary, the following columns would be needed:

- agreement_no - - - - - > mapped as CASE_ID
- primary_occupation
- primoccup_group
- region_name
- region_group
- joint_total_income
- amount_drawn
- loan_term
- primary_date_of_birth
- completion_date
- first_val_date
- loan_purpose
- purchase_price
- primary_self_employed

- portf_pool
- current_default_flag
- app_scorecard_PD
- PD
- pool3

The first column (agreement_no) is the Primary Key, which identifies each record as a unique record. Even though this is just an ID column, to fully ensure to qualify for data privacy, the data in this column should be replaced with a self-generated identifier, which could be called: CASE_ID. The column (portf_pool) indicates whether an application performs, defaults or is irrecoverable. This column will be used for the verification how the model performed. The name for the column (current_default_flag) is a bit misleading, because this column shows whether an application is or has been defaulted, i.e. in arrears or not, but this column is not constantly updated, so in total it contains both the present and past, so this only indicates whether a mortgage application ever had a difficulty and not the actual live status. The columns (app_scorecard_PD , PD , pool3) are the result components of the scorecard method. All other columns (primary_occupation , primoccup_group , region_name , region_group , joint_total_income , amount_drawn , loan_term , primary_date_of_birth , completion_date , first_val_date , loan_purpose , purchase_price , primary_self_employed) are used for the calculation of the arguments for the computational AT approach.

In conclusion of the data understanding, it was determined, that the dataset which was formed contained all the necessary information for the experiment execution.As per Hypothesis H0, the Scorecard Method, as described in the section Business Understanding (2.2.2) is taken into consideration as baseline. However, even if the Scorecard Method's formulas have been defined as per equations (2.1 and 2.2), the exact input details for such formulas have not always been provided by the bank, because of privacy and anonymity reasons.

The dataset is to be ingested into SQL server 2014 Community Edition (this edition is a free software) and for simplicity it is recommended to use varchar(50) for every column, which means string 50 characters long. All columns were double checked for this size and this operation can be done without any data truncation and without any data loss.

There was a data quality check performed on the second (main) data set by selecting 500 records as a sample extract. This sample data set was prepared using the above mentioned software: SQL Server. Before ingesting the sample data into the software, the data had to be converted from string to the proper format to be able to qualify for data quality check operations, so to be able to execute mathematical functions. The input features (columns) for the data quality report are those 14 features, which are needed for the risk of defaulting calculations. A Data Quality Report is produced for the mortgage applications dataset as per following process. The report can be categorized into two groups, based on two different set of features: whether the data is a numeric field (continuous), or if it is a text field (categorical).

(a) Continuous Features:

- Feature 1 - primoccup_group - numeric
- Feature 2 - region_group - numeric
- Feature 3 - joint_total_income - numeric (money)
- Feature 4 - amount_drawn - numeric (money)
- Feature 5 - loan_term - numeric
- Feature 6 - primary_date_of_birth - numeric (date)
- Feature 7 - completion_date - numeric (date)
- Feature 8 - first_val_date - numeric (date)
- Feature 9 - purchase_price - numeric (money)

The sample dataset for the continuous features was prepared using SQL code as per referenced table in the Appendix D.1.

After the data was extracted from SQL server as a CSV file, it was ingested into R-Studio. The numerical analysis was done using R code as per referenced table in the Appendix D.2.

The result of the DQ (Data Quality) analysis, for the continuous features is as per following tables 4.1 and 4.2 and 4.3 :

Feature	**Count**	**Missing**	**Unique**
primoccup_group	500	0	5
region_group	500	0	4
joint_total_income	500	2	484
amount_drawn	500	0	278
loan_term	500	0	150
primary_date_of_birth	500	0	487
completion_date	500	10	259
first_val_date	500	2	276
purchase_price	500	3	257

Table 4.1: The Data Quality Table - Continuous Features I

Feature	**Min**	**Mean**	**Max**
primoccup_group	1.000	2.708	5.000
region_group	1.00	1.54	4.00
joint_total_income	0	112544	865910
amount_drawn	0	276134	1296000
loan_term	2.12	25.98	37.96
primary_date_of_birth	1948-04-15	1977-08-05	1990-03-25
completion_date	1999-11-02	2014-10-14	2016-12-19
first_val_date	1999-09-20	2014-08-06	2017-01-20
purchase_price	0	375916	4410000

Table 4.2: The Data Quality Table - Continuous Features II

Feature	**1st Qrt.**	**Median**	**3rd Qrt.**
primoccup_group	2.000	3.000	4.000
region_group	1.00	1.00	2.00
joint_total_income	73255	94600	125072
amount_drawn	175000	244350	340000
loan_term	23.02	28.06	30.03
primary_date_of_birth	1974-04-07	1978-10-17	1982-01-17
completion_date	2014-08-27	2014-11-11	2015-03-24
first_val_date	2014-06-23	2014-08-13	2014-12-10
purchase_price	245000	335000	455000

Table 4.3: The Data Quality Table - Continuous Features III

(b) Categorical Features:

- Feature 10 - agreement_no - string, primary key (unique value, mandatory field, listed for completeness, but ignored for data quality report)

- Feature 11 - primary_occupation - string

- Feature 12 - region_name - string
- Feature 13 - loan_purpose - string
- Feature 14 - primary_self_employed - string

The sample dataset for the categorical features was prepared using SQL code as per referenced table in the Appendix D.3. The DQ (Data Quality) analysis, for the categorical features, were performed using individual SQL codes, feature by feature (using WHERE conditions to check missing values, DISTINCT operation to check unique values, ORDER BY and derived columns to calculate mode frequency and mode percentages) and the following results were gathered manually into the forthcoming tables 4.4 and 4.5 and 4.6 :

Feature	**Count**	**Missing**	**Unique**
primary_occupation	500	0	82
region_name	500	0	43
loan_purpose	500	0	6
primary_self_employed	500	0	2

Table 4.4: The Data Quality Table - Categorical Features I

Feature	**Mode**	**Mode Freq.**	**Mode %**
primary_occupation	It Consultant	48	9.6
region_name	CO DUBLIN	80	16.0
loan_purpose	House Purchase mover	236	47.2
primary_self_employed	E	468	93.6

Table 4.5: The Data Quality Table - Categorical Features II

Feature	**2nd Mode**	**2nd Mode Freq**	**2nd Mode %**
primary_occupation	Manager General	39	7.8
region_name	CO CORK	45	9.0
loan_purpose	House Purchase First Time Buyer	210	42.0
primary_self_employed	S	32	6.4

Table 4.6: The Data Quality Table - Categorical Features III

Reminder: it is important to call out on the results above, that this is not the analysis for the full dataset, but only for the 500 records sample, as a representative sample, for data quality checking purposes.

The data has to be understood, analysed and interpreted. First, it has to be determined how well the scorecard method predicted, as that is the benchmark. The dataset contains a column which indicates whether a case is performing well, or defaulting or if it is irrecoverable. The irrecoverable can be also considered as a form of long term defaulting. The following chart 4.1 visualizes this result, with these mentioned categories and displays the amount of applications and also as a percentage:

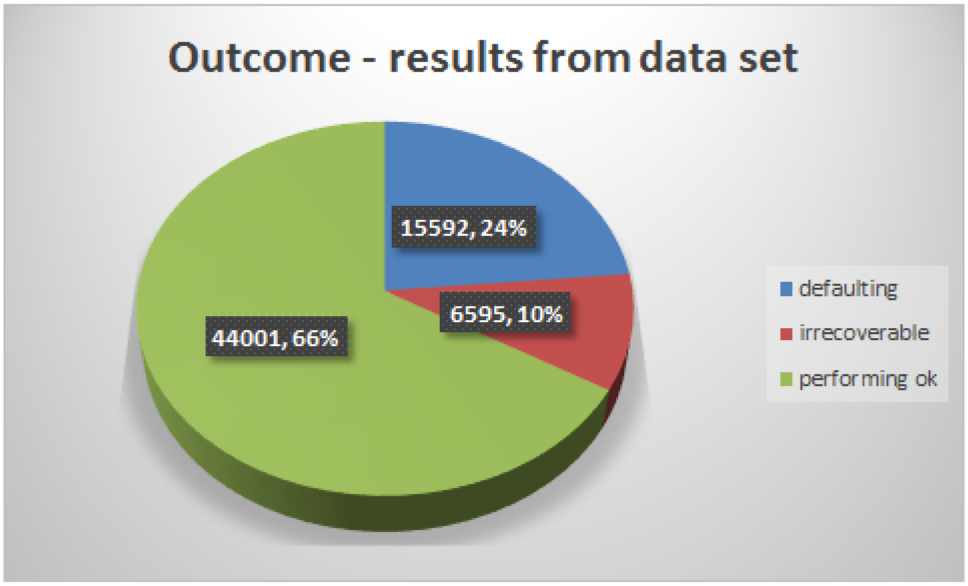

Figure 4.1: Outcome - result based on the data set (version i)

Strictly speaking the business is much more concerned about the irrecoverable cases only, which is 10%, as those have immediate financial consequences. Unless of course all those 24% defaulting cases will turn to irrecoverable, but in the reality that is usually not the case. Most of the time those cases are defaulting only temporarily. that is the job of the Arrears Support Unit to chase these 24% cases and so-called cure them, which means taking back a non-functioning mortgage to the functioning route. This usually happens via restructuring the mortgage. In this sense it means, that the scorecard method predicted well in 90% of the cases and did not predict well in 10% of the cases. The visual representations of these results can be seen on the following pie chart 4.2:

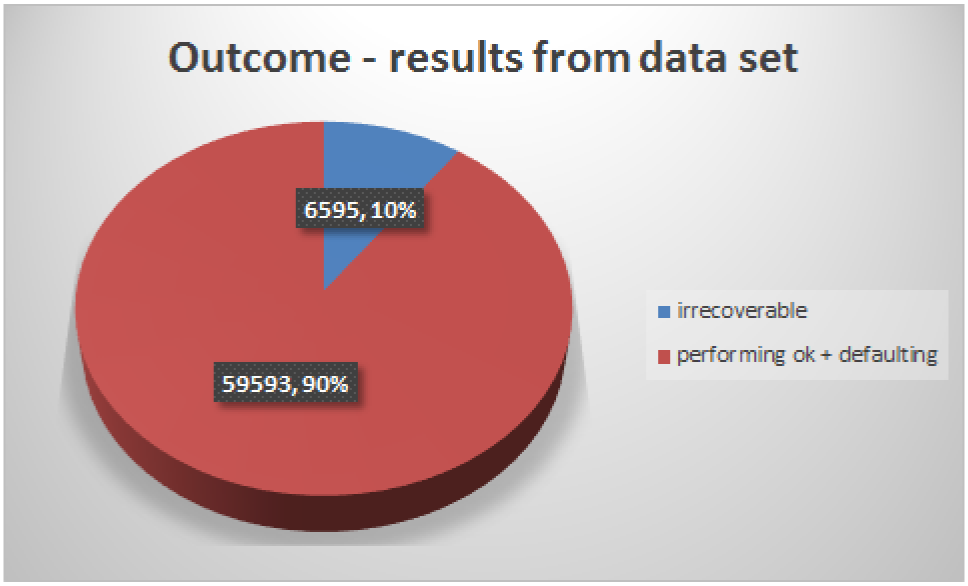

Figure 4.2: Outcome - result based on the data set (version ii)

This is only pure speculation, but if we use the same ratios of the data set to the defaulting cases only, i.e. 10% goes to irrecoverable then the 10% of the 24% is 2.4% which will turn to irrecoverable, which is 1583 cases (= 6595 / 10 * 2.4). This figure may not be a large number compared to the whole lot, but still a 1583 families. This could be interesting as a future work possibility to follow up the cases and check the irrecoverable ratio statistics for these cases in the future.

It is possible to draw the following confusion-matrix from the same data 4.7:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 6 595 (10% +)	N/A
Actual: YES	FN = N/A	TP = 59 593 (90% +)	N/A
	N/A	66 188 (100% +)	

Table 4.7: confusion-matrix - scorecard method I

Some of the abbreviations in the table above are definitions and require clarifications. Namely:

- TP = True Positives: meaning those cases which predicted YES (GOOD) - not defaulting in our case and they are really YES (GOOD) cases and performing well

- TN = True Negatives: those cases where the prediction is NO (BAD) - defaulting and they are really NO (BAD) defaulting mortgage applications
- FP = False Positives: are those cases where the prediction is YES (GOOD) - not defaulting, but they are actually NO (BAD) cases in reality, and actually they are in arrears. This one is also known as Type I error.
- FN = False Negatives: happen when the prediction is NO (BAD) - defaulting, but they are actually not bad cases, they are actually YES (GOOD) cases and not defaulting. This is also called as Type II error.

As per the confusion-matrix above 4.7 the left side of the matrix is completely missing. The n = N/A also represents the same, as the total records of the data set is not known, only the accepted cases, so only those applicants who actually got the mortgage. **This is the biggest limitation of this study and the data set.** This is by definition so, because those cases where the prediction is NO are automatically rejected by the bank, as the application is too risky to proceed. It would be extremely rarely that the Underwriting Department who is actually recommending the decision would override the decision of the Risk Department on this matter.

Another remark is, that there is a + sign beside the percentage values, the reason for that is, that actually by definition the 100% supposed to be a higher number, which right now only represents the right side of the confusion-matrix, as the left side is missing. The reason to show that as 100% is for representative purposes, to be able to relate to the chart 4.2.

The other scenario is, if we are trying to predict the ratio of the problematic cases. In this sense the irrecoverable and the defaulting cases should be counted together, which is 34%, as both are problematic and require further actions from the bank. This is supported by the ultimate goal of the bank's Risk Department which is to predict who will have difficulties in paying back the mortgage and maximise the bank's profit via minimizing the risk. In other words, in this scenario, 66% of the cases (or 2/3rd) the bank's scorecard method predicted well and 34% of the cases the scorecard did not predict well. This can be represented in the following chart 4.3:

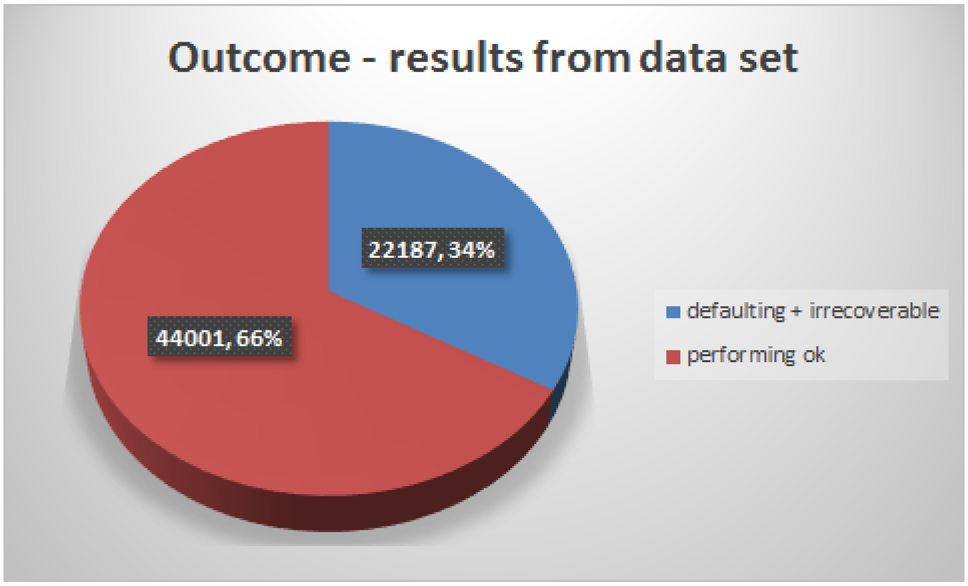

Figure 4.3: Outcome - result based on the data set (version iii)

It is also pssible to draw the following confusion-matrix from the same data as per following table 4.8:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 22 187 (34% +)	N/A
Actual: YES	FN = N/A	TP = 44 001 (66% +)	N/A
	N/A	66 188 (100% +)	

Table 4.8: confusion-matrix - scorecard method II

Important remark, the + sign beside the percentage values at the confusion-matrix 4.8 above mean that only the right side of the matrix is available, the left side is N/A - meaning not available.

4.2 Layer 2. - Data preparation

For DR the exact same dataset is in use as originally used for the scorecard method by the medium size Irish bank. The difference is, that for DR only selected columns were used, those which are essential to build the knowledege base, so only a sub-set of columns were taken.

Two data sets were received from the bank and there is a big overlay between the 2 data sets, namely 6382 records out of 6535 records, which means that 97.7% of the first data set also can be found in the second data set, so as a conclusion of this, the first data set was emitted and only the second data set was used to execute the final experiment.

To begin with the data preparation a handful of duplicated rows were eliminated from the data set, exactly 8 rows, using the DISTINCT method in SQL. As mentioned in the Design chapter, the first dataset has 6535 records and the second dataset has 66 196 records. It was discovered as part of the initial data quality checks, that there are a handful of duplicated records in the second (the main) data set. After using a SELECT DISTINCT command on all columns, 66 188 unique records remain.

As the next step, the empty strings were replaced with NULL values, for all columns, to be able to use a larger number of available functions, procedures and methods.

As it is possible to see in the Data Quality (DQ) Report, there are some values missing in the data which requre attention. The next step is to resolve the DQ issues, where ever it is possible. Among the critical features, the Date Of Birth of the primary applicant requires some work, because there are 18 cases without any traceable values and without any alternative feature. For those set of arguments which uses this feature the high risk value needs to be assigned as a default value for the result of the prediction of that particular argument. There are 436 affected cases without region_name out of 66188 records, which is 0.7% , negligible, but to facilitate a better prediction, the missing values were treated with a replacement value, using [region_group] as an alternative available column.Among the critical numerical features which have a data quality problem, the [joint_total_income] has no value 149 times and takes the zero (0) value another 637 times. To overcome this issue the missing values have to be considered as zero to be able to perform mathematical operation, but it has to be ensured to avoid the divide by zero error. The logic for these cases has to ensure that if this feature is used for an argument then the argument should predict the highest risk value, due to the missing data. a further case the [completion_date] is missing for 740 records, but there is an alternative way to fill in this value. The feature [first_val_date] can be used as a replacement of [completion_date] as a workaround solution, because these dates in reality should be very close to each other. Where the [purchase_price] is not filled, a significant 5298 records, and an additional 21604 cases where the value is set to zero, and the two together represent 41% of the total data set! even though the calculation will not be so accurate, but due to the high number of error an alternative is essential so the [amount_drawn] feature is in use instead. This reduces the error to 90 cases where the [amount_drawn] equals zero.

4.3 Layer 3. - Internal structure of arguments

The arguments were defined following the conceptual model:

i) Forecasting arguments for acceptance

ii) Forecasting arguments for rejection

iii) Undercutting arguments

The implementation details are in the following sections.

4.3.1 Forecasting arguments for acceptance

There were 7 initial sets of arguments defined as described with pseudo code as per the following tables:

1st set of Arguments:

Arg. label	**Arg. definition**
R1_L	CASE WHEN PrimaryOccupation IN (Occup a1, Occup a2, . . ., Occup an) THEN Low Risk
R1_ML	CASE WHEN PrimaryOccupation IN (Occup b1, Occup b2, . . ., Occup bn) THEN Medium-Low Risk
R1_M	CASE WHEN PrimaryOccupation IN (Occup c1, Occup c2, . . . , Occup cn) THEN Medium Risk
R1_MH	CASE WHEN PrimaryOccupation IN (Occup d1, Occup d2, . . ., Occup dn) THEN Medium-High Risk

Table 4.9: The Forecasting Arguments table - supporting acceptance - F. A. 1

For this set of arguments, a list of 237 occupations will be split and put into 5 different buckets, based on the probability of risk, whether it is a safe job, or if it is a risky job.

2nd set of Arguments:

Arg. label	**Arg. definition**
R2_L	CASE WHEN Region IN ('Dublin') THEN Low Risk
R2_ML	Argument 2 - only used as a numerical value when the text value is missing
R2_M	CASE WHEN Region IN (Other towns than Dublin and neighbour counties of Co. Dublin, i.e. Co. Meath, Co. Kildare, Co. Wicklow; and additionally Co. Cork and Co. Galway) THEN Medium Risk
R2_MH	Argument 4 - only used as numerical value when text value is missing

Table 4.10: The Forecasting Arguments table - supporting acceptance - F. A. 2

These set of arguments are categorizing Ireland (and the rest of the world as part of everything else) whether it is a risky location to invest in a property or whether it is not. The general rule is, if it is a popular and populated area than it is less risky, so the starting point is the capital of Ireland, Dublin, and that is considered as the safest area to invest in property, via mortgages.

The following comment in the table would require a short explanation: *only used as a numerical value when the text value is missing* this is a recommendation, that even though generally the column [region_name] is used for the calculation of the argument, which can be taken directly from the source data, but that is only possible if that value is available. If that value is missing, which is the case only in 0.7% of the cases, then the recommendation is to use an alternative value. In this particular case, the source data contains another region categorization field as a numerical value, which is called [region_group]. It was double checked, that the value in this field is never blank.

3rd set of Arguments:

Arg. label	**Arg. definition**
R3_L	CASE WHEN Yearly_Income * 4.4 > LoanAmount THEN Low Risk
R3_ML	Argument 2 - not in use
R3_M	Argument 3 - not in use
R3_MH	CASE WHEN Yearly_Income * 4.4 > LoanAmount > Yearly_Income * 3.5 THEN Medium-High Risk

Table 4.11: The Forecasting Arguments table - supporting acceptance - F. A. 3

These set of arguments are focusing on the relationship between the Income and the Loan amount. The general rule here should be, that the higher the income percentage is compared to the loan amount, the better the chances are, that the applicants can pay back their mortgage without any difficulties.

4th set of Arguments:

Arg. label	**Arg. definition**
R4_L	IF MaxLoanTerm <= 35 Years THEN GoodLoan (Low Risk)
R4_ML	Argument 2 - not in use
R4_M	Argument 3 - not in use
R4_MH	Argument 4 - not in use

Table 4.12: The Forecasting Arguments table - supporting acceptance - F. A. 4

These set of arguments are focusing solely on the maximum period of the loan term. Initially, this should be capped at 35 years.

5th set of Arguments:

Arg. label	**Arg. definition**
R5_L	IF (Age <= 30 AND Age + LoanTerm <= 68) OR (Age > 30 AND Age + LoanTerm <= 65) THEN GoodLoan (Low Risk)
R5_ML	Argument 2 - not in use
R5_M	Argument 3 - not in use
R5_MH	Argument 4 - not in use

Table 4.13: The Forecasting Arguments table - supporting acceptance - F. A. 5

These set of arguments are examining the relationship between the Loan Term and the Age of the primary applicant, at Maturity Date i.e. the date when the mortgage is paid back in full. A distinguishing is recommended for an age group of under 30 years of age, as for those people a longer Loan Term should be allowed. The rationale behind these set of arguments are based on the demographical statistics in the country of Ireland.

6th set of Arguments:

Arg. label	Arg. definition
R6_L	CASE WHEN RepaymentYearly * 2.0 > NDI THEN Low Risk
R6_ML	CASE WHEN RepaymentYearly * 1.6 > NDI > RepaymentYearly * 2.0 THEN Medium-Low Risk
R6_M	CASE WHEN RepaymentYearly * 1.4 > NDI > RepaymentYearly * 1.4 THEN Medium Risk
R6_MH	CASE WHEN RepaymentYearly * 1.2 > NDI > RepaymentYearly * 1.0 THEN Medium-High Risk

Table 4.14: The Forecasting Arguments table - supporting acceptance - F. A. 6

These set of arguments are comparing the Yearly Loan Repayment values versus the NDI (Net Disposable Income). The general expectation is: the higher the repayment percentage is the less money is available from the Net Disposable Income, so higher repayment percentage would be a higher risk for paying back the mortgage.

To calculate the NDI (Net Disposable Income) the following simplified formula can be used:
CASE WHEN YearlyIncome < 30k THEN NDI = 0
CASE WHEN 30k < YearlyIncome < 40k THEN NDI = YearlyIncome * 0.3
CASE WHEN 40k < YearlyIncome < 60k THEN NDI = YearlyIncome * 0.4
CASE WHEN YearlyIncome > 60k THEN NDI = YearlyIncome * 0.5
This calculation is essential, because the NDI is not directly part of the data set, but can be calculated indirectly, using the logic above, based on the total income.

To calculate the RepaymentYearly value, as that is not part of the data set directly either, the following calculation could be used. It can be derived from the combination of the loan amount drawn and the loan term:
To calculate the total PaybackAmount = (Loan * ((MarketInterest% / 2) POWER LoanTerm))).
For simplicity it is possible to use a market interest rate percentage of 4 %. This 4% equals to = 0.04 then to calculate the Interest% = MarketInterest% / 2 = 0.02 .

For better understanding, see an illustration of these additional calculations with an Example here:

PaybackAmount = 100k * (1.02 POWER 30) = 100k * 1.81 = 181k.

LoanTerm is an input of: 5 . . . 35 years, let's pick for example: 30 years.

RepaymentYearly = PaybackAmount / LoanTerm

7k = 200k / 30 years

These calculations could be reviewed as future work possibilities, to define more accurate conditions for the several inputs of the Arguments, like NDI or PaybackAmount.

As another example, for example for KBC, a real PaybackAmount calculator can be found here on this url:

(https://www.kbc.ie/Our-Products/Mortgages/How-much-can-I-borrow)

7th set of Arguments:

These set of arguments should be split into two sub-sets, based on whether the applicants are First Time Buyers or not.

IF FirstTimeBuyer = True THEN:

Arg. label	**Arg. definition**
R7_L	CASE WHEN (Loan / Value of House) < 0.6 THEN Low Risk
R7_ML	CASE WHEN 0.6 < (Loan / Value of House) < 0.7 THEN Medium-Low Risk
R7_M	CASE WHEN 0.7 < (Loan / Value of House) < 0.8 THEN Medium Risk
R7_MH	CASE WHEN 0.8 < (Loan / Value of House) < 0.9 THEN Medium-High Risk

Table 4.15: The Forecasting Arguments table - supporting acceptance - F. A. 7a

IF FirstTimeBuyer = False THEN:

Arg. label	**Arg. definition**
R7_L	CASE WHEN (Loan / Value of House) < 0.5 THEN Low Risk
R7_ML	CASE WHEN 0.5 < (Loan / Value of House) < 0.6 THEN Medium-Low Risk
R7_M	CASE WHEN 0.6 < (Loan / Value of House) < 0.7 THEN Medium Risk
R7_MH	CASE WHEN 0.7 < (Loan / Value of House) < 0.8 THEN Medium-High Risk

Table 4.16: The Forecasting Arguments table - supporting acceptance - F. A. 7b

These set of arguments compare the loan with the value of the property and recommend to use different ranges for the first time buyers and for those applicants, who are not first time buyers. The general expectation is, that the smaller the percentage of the loan is compared to the value of the house, the better, and as a consequence, less risk is involved.

4.3.2 Forecasting arguments - supporting rejection

The forecasting arguments which are supporting rejection are described with pseudo code as per following.

Argument 1:

Arg. label	**Arg. definition**
R1_H	CASE WHEN PrimaryOccupation IN (Occup e1, Occup e2, . . ., Occup en) THEN High Risk

Table 4.17: The Forecasting Arguments table - supporting rejection - A. 1

If the occupation is part of a specific selected occupations list, which would bear high risk, that means, those occupations would not be favourable.

Argument 2:

Arg. label	Arg. definition
R2_H	CASE WHEN Region IN (Everything else than: 'Dublin' and Other towns than Dublin and neighbour counties of Co. Dublin, i.e. Co. Meath, Co. Kildare, Co. Wicklow; and Co. Cork and Co. Galway) THEN High Risk

Table 4.18: The Forecasting Arguments table - supporting rejection - A. 2

The location would be an important factor, high risk would be considered if it is not easy to market the property or if the property is likely lose from it's value.

Argument 3:

Arg. label	Arg. definition
R3_H	CASE WHEN Yearly_Income * 3.5 < LoanAmount THEN High Risk

Table 4.19: The Forecasting Arguments table - supporting rejection - A. 3

If the income is lower than a certain threshold, or from another aspect, if the applicant would like to have an unreasonable high amount of loan, then the applicant can't afford to pay back the mortgage, so the risk would be very high to proceed with the granting the loan.

Argument 4:

Arg. label	Arg. definition
R4_H	IF MaxLoanTerm > 35 Years THEN BadLoan (High Risk)

Table 4.20: The Forecasting Arguments table - supporting rejection - A. 4

The longer the loan term is, the lower will be the total yearly repayment amount, which makes it more affordable, however, the total cost of the mortgage will be increase and this can not go on to unlimited lengths.

Argument 5:

Arg. label	**Arg. definition**
R5_H	IF (Age <= 30 AND Age + LoanTerm > 68) OR (Age > 30 AND Age + LoanTerm > 65) THEN BadLoan (High Risk)

Table 4.21: The Forecasting Arguments table - supporting rejection - A. 5

The older an applicant will be at the final payback date of the mortgage, the higher is the risk.

Argument 6:

Arg. label	**Arg. definition**
R6_H	CASE WHEN RepaymentYearly * 1 < NDI THEN HighRisk

Table 4.22: The Forecasting Arguments table - supporting rejection - A. 6

Yearly repayments should be affordable by the applicant. If the mortgage would take up most of the income leftovers, or if the expenses would even exceed the income, then paying back the mortgage would not be possible and this would indicate a high risk case.

Argument 7:

Two sub-arguments should be distinguished as per following:

IF FirstTimeBuyer = True THEN:

Arg. label	**Arg. definition**
R7_H	CASE WHEN (Loan / Value of House) > 0.9 THEN High Risk

Table 4.23: The Forecasting Arguments table - supporting rejection - A. 7a

IF FirstTimeBuyer = False THEN:

Arg. label	**Arg. definition**
R7_H	CASE WHEN (Loan / Value of House) > 0.8 THEN High Risk

Table 4.24: The Forecasting Arguments table - supporting rejection - A. 7b

If the loan amount is nearly as much as the value of the house, that would represent a high risk, because the applicant should be able to present some available capital to prove the ability to save up for the repayments.

4.3.3 Undercutting arguments

The exception arguments (undercutting arguments) are as per following:

To Argument 1 :

Arg. label	**Arg. definition**
R8_MH	If Occupation is in (A , ...) AND Permanent contract and earning above XX k THEN Lower risk to Medium-High

Table 4.25: The Undercutting Arguments table - U. A. 1

Even though the occupation is listed as high risk, but if the applicant is an employee and earning above a threshold then this case could be considered as not a high risk case anymore.

To Argument 2:

Arg. label	**Arg. definition**
R9_MH	CASE When Region IN (HighRisk) but Permanent contract and earning above XXk THEN Lower risk to Medium

Table 4.26: The Undercutting Arguments table - U. A. 2

If the property is on a not preferred location, but the applicant is earning more than a certain sum of money and is in permanent employment, then the risk would not be that significant as originally indicated.

To Argument 3:

Arg. label	**Arg. definition**
R10_MH	CASE WHEN Yearly_Income * 3.5 > LoanAmount > Yearly_Income * 3.0 AND Occupation risk is between 1..3 THEN Medium-High Risk

Table 4.27: The Undercutting Arguments table - U. A. 3

In case the Loan versus income ratio in itself is not ideal, but it is still within a manageable limits and the applicant has a favourable occupation, then that can be considered as a lower risk level than originally calculated.

To Argument 4:

Arg. label	**Arg. definition**
R11_MH	CASE WHEN 35 Years < MaxLoanTerm < 40 Years THEN Lower Risk to Medium-High

Table 4.28: The Undercutting Arguments table - U. A. 4

When the loan term is just outside the specified limits, then this application could be considered as not as risky as initially indicated.

To Argument 5:

Arg. label	**Arg. definition**
R12_MH	CASE WHEN 65 Years < Age + LoanTerm < 70 THEN Reduce risk to Medium-High

Table 4.29: The Undercutting Arguments table - U. A. 5

For the expected age at Maturity Date, at final payback, if the age is just above the nominated limit, then this case still can be considered, exceptionally, as a lower risk than high risk.

To Argument 6:

Arg. label	**Arg. definition**
R13_MH	CASE WHEN RepaymentYearly * 0.9 < NDI < RepaymentYearly * 1.0 THEN reduce risk to Medium-High Risk

Table 4.30: The Undercutting Arguments table - U. A. 6

When the affordability is just on the limit, the applicants, exceptionally, still could be considered as reliable, who could pay back their mortgage, as the NDI is a relative complex calculation and also has some built-in reserve amounts.

To Argument 7:

Two sub-sets of arguments:

IF First_time_buyer = True :

Arg. label	**Arg. definition**
R14_MH	CASE WHEN 0.9 < (Loan / Value of House) < 0.95 THEN reduce risk to Medium-High Risk

Table 4.31: The Undercutting Arguments table - U. A. 7a

IF First_time_buyer = False :

Arg. label	**Arg. definition**
R14_MH	CASE WHEN 0.8 < (Loan / Value of House) < 0.9 THEN reduce risk to Medium-High Risk

Table 4.32: The Undercutting Arguments table - U. A. 7b

Depending on whether the applicant is a first time buyer or not, different ranges should apply, but similar concept, if the loan is nearly as much as the value of the property, but it is still within a very reasonable range, then the risk could be treated as lower than a high risk case, exceptionally.

To summarize this layer, three sets of arguments were created, forecaseting arguments supporting acceptance, forecasting arguments supporting rejection and undercutting arguments. The definition of the arguments are described with pseudo code, but for the actual implementation into SQL Server the arguments definition were slightly adjusted, to be able to better accommodate the pseudo code language to the SQL code language. The exact details which were used can be seen at the Appendix A. There were no error messages when compiling the SQL code, every single argument could be executed and they all return a result, which could be used for further calculations in the forthcoming processes. To execute the experiment a web ui frontend interface was used to model the mortgage assessment with computational AT. The sets of arguments (nodes) were all created there. As the knowledge base contains complex calculations, SQL language was used to pre-process and generate a numerical value for each argument. The value of the argument is a decimal value between 1 (- best) and 5 (- worst). This value was later ingested as the input value for the argument, for further calculations of the computational AT approach. The relationships between the arguments also had to be created (which argument is attacking which other argument), as per the following section.

4.4 Layer 4. - Conflicts of arguments

The attacks were defined following the conceptual model, as per design chapter:

i) the set of forecasting arguments for acceptance are not attacking anything; but they are all attacked by the forecasting arguments for rejection;

ii) The undercutting arguments are attacking the forecasting arguments for rejection; and the undercutting arguments are also attacked by all those other forecasting arguments for rejection which are not attacked by the particular undercutting argument.

The conflicts themselves have no immediate values, as binary attacks were used (either the attack exist, or virtually does not exist), hence the conflicts are used as part of the rules, for forthcoming calculations.

At this layer the arguments and the relationship between the arguments were established, the argument graph is ready. However, it has to be established which argument is winning or losing over the other argument and these details are described in the next sections.

4.5 Layer 5. - Dialectical status of arguments

Based on the calculation of the participating arguments and the direction of the attacks, the evaluation of the arguments start on the outside, at that argument in the argument graph, which is not attacked. Initially those arguments will be labelled as IN. In the next step, those arguments, which are attacked by the IN labelled argument will be labelled as OUT. To continue the cycle, because this argument is now labelled as OUT, all those arguments which were attacked by this argument will be labelled now as IN. This cycle continues until every argument is labelled either IN or OUT. At the end of the cycle, if there are still some arguments, which do not have a label, those will be labelled as UNDEC. Those arguments, which are not relevant for the calculation are simply not taken into consideration.

After executing this phase, all relevant arguments are labelled IN or OUT or UNDEC, for each case, i.e. for each mortgage application. The following step is to calculate the PD score, as per next section.

4.6 Layer 6. - Accrual of acceptable arguments

The calculation of an example of loan defaulting modeled with Computational AT (Argumentation Theory) is as per following. First the SQL rules had to be executed in a different environment to calculate the arguments which formed the input for the AT web application. Once the SQL calculation was done, then the following table was be exported out of SQL Server 4.4:

CaseID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
B164553	1	1	1	1	4	1	3	1	1	1	1	4	1	3

Figure 4.4: Computational AT example - input file with the arguments

The table above 4.4 was extracted in CSV format (Coma Separated Values) and was uploaded then to the AT application, as an input file. The AT application had to be setup with the modeling details of the argumentation and then the application calculated and evaluated the arguments and computed the final results, based on the selected semantics, which were grounded and preferred semantics. Other possible semantics for the tool would have been: Expert, Eager, Ideal, Stable, Semi-stable, Admissible, but those do not conform with calculating risk assessment. The results were exported from the AT UI platform. For this study the mean Average was used to calculate the final PD score for AT within those arguments which were labelled IN. Other methods on the WEB UI were possible like Sum, Highest cardinality, Median and Weighted average, but for simplicity and easy understanding Average was selected. The following table 4.33 is showing the outcome :

CaseID	Grounded semantics	Preferred semantics
B164553	1.71	1.71

Table 4.33: Computational AT example - results

In this particular case, the calculation summary result, as the outcome of the argumentation is: 1.71, for both applicable semantics. This result is a representation of the forecasting arguments which support acceptance. This time those arguments which support rejection were not taken into consideration for the final calculation, as this is a low risk of PD. The evaluation of the final score of 1.71, in this particular case, qualifies as a good result. That means that this mortgage application probably will not default. The scale range which was set for this study for computational AT ranges from 1 to 5, where the good results range from 1 to 3. In other words, those applicants who get a score between 1 and 3, they probably will not have any problems paying back their mortgage, and therefore it is unlikely that they will default and run into mortgage re-payment difficulties.

To summarize the results for computational AT there are 2 applicable semantics for this study: grounded semantics and preferred semantics.

(a) Grounded semantics

Scenario I - confusion-matrix - for cases when only irrecoverable is the bad case 4.34:

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 3 446 (5%)	FP = 3 149 (5%)	6 595 (10%)
Actual: YES	FN = 25 168 (38%)	TP = 34 425 (52%)	59 593 (90%)
	28 614 (43%)	37 574 (57%)	

Table 4.34: confusion-matrix - computational AT irrecoverable (version i)

By looking at the results above, for Scenario I, the summary result is the TN + TP = 5% + 52% = 57% meaning this is where AT predicted well and FN + FP = 38% + 5% = 43% where AT did not predict well. Compared to flipping a coin, where the probability is 50% this is still a +7% better result.

To be able to do the comparison, let's calculate only the right side as a total of 100%, this can be seen in the following matrix 4.35:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 3 149 (9% +)	N/A
Actual: YES	FN = N/A	TP = 34 425 (91% +)	N/A
	N/A	37 574 (100% +)	

Table 4.35: confusion-matrix - computational AT irrecoverable (version ii)

Scenario II - confusion-matrix - for cases when irrecoverable and defaulting together are the bad cases 4.36:

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 11 060 (17%)	FP = 17 554 (26%)	28 614 (43%)
Actual: YES	FN = 11 127 (17%)	TP = 26 447 (40%)	37 574 (57%)
	22 187 (34%)	44 001 (66%)	

Table 4.36: confusion-matrix - computational AT irrecoverable + defaulting (version i)

To be able to do the comparison, let's calculate only the right side again as a total of 100%, the result is as per following matrix 4.37:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 17 554 (40% +)	N/A
Actual: YES	FN = N/A	TP = 26 447 (60% +)	N/A
	N/A	37 574 (100% +)	

Table 4.37: confusion-matrix - computational AT irrecoverable + defaulting (version ii)

(b) Preferred semantics

After the analysis of the results, the difference between Grounded semantics and Preferred semantics is that the Preferred semantics not returning any results for 7 109 cases, which is 13% of the total cases, due to unresolvable route in the calculation.

The other 87% of the cases the Grounded and Preferred semantics return the exact same result. These 13% NULL cases need to be categorized, so the result calculations can be concluded. There are 3 ways of doing so. Evaluate the results via:

A) disregarding the NULL cases and by doing so reduce the data set size

B) consider those NULL cases as high risk, i.e. consider those cases as they are defaulting

C) consider those NULL cases as low risk, i. consider those cases as they are not defaulting

Each of these A) and B) and C) can have the following 2 scenarios:

Scenario I: only irrecoverable is the bad case

Scenario II: irrecoverable and defaulting are together the bad cases

So there are going to be 6 different results and each of these 6 is also re-calculated to calculate the right side of the confusion-matrix only, so the final result set for preferred semantics will be represented in the following 12 tables.

A) NULL is disregarded - Scenario I - confusion-matrix - for cases when only irrecoverable is the bad case 4.38 :

n = 59 079	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 3 119 (5%)	FP = 2 764 (5%)	5 883 (10%)
Actual: YES	FN = 22 366 (38%)	TP = 30 830 (52%)	53 196 (90%)
	25 485 (43%)	33 594 (57%)	

Table 4.38: confusion-matrix - computational AT irrecoverable - A) - version i

To be able to do the comparison, let's calculate only the right side as a total of 100%, this can be seen in the following matrix 4.39:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 2 764 (8% +)	N/A
Actual: YES	FN = N/A	TP = 30 830 (92% +)	N/A
	N/A	33 594 (100% +)	

Table 4.39: confusion-matrix - computational AT irrecoverable - A) - version ii

A) NULL is disregarded - Scenario II - confusion-matrix - for cases when irrecoverable and defaulting together are the bad cases 4.40:

n = 59 079	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 11 285 (17%)	FP = 9 847 (17%)	20 132 (34%)
Actual: YES	FN = 15 200 (26%)	TP = 23 747 (40%)	38 947 (66%)
	25 485 (43%)	33 594 (57%)	

Table 4.40: confusion-matrix - computational AT irrecoverable + defaulting - A) - version i

To be able to do the comparison, let's calculate only the right side again as a total of 100%, the result is as per following matrix 4.41:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 9 847 (29% +)	N/A
Actual: YES	FN = N/A	TP = 23 747 (71% +)	N/A
	N/A	33 594 (100% +)	

Table 4.41: confusion-matrix - computational AT irrecoverable + defaulting - A) version ii

B) NULL is high risk - Scenario I - confusion-matrix - for cases when only irrecoverable is the bad case 4.42 :

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 3 119 (5%)	FP = 2 764 (4%)	5 883 (9%)
Actual: YES	FN = 29 475 (45%)	TP = 30 830 (47%)	60 305 (91%)
	32 594 (49%)	33 594 (51%)	

Table 4.42: confusion-matrix - computational AT irrecoverable - B) - version i

To be able to do the comparison, let's calculate only the right side as a total of 100%, this can be seen in the following matrix 4.43:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 2 764 (9% +)	N/A
Actual: YES	FN = N/A	TP = 30 830 (91% +)	N/A
	N/A	33 594 (100% +)	

Table 4.43: confusion-matrix - computational AT irrecoverable - B) - version ii

B) NULL is high risk - Scenario II - confusion-matrix - for cases when irrecoverable and defaulting together are the bad cases 4.44:

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 17 394 (26%)	FP = 9 847 (15%)	27 241 (41%)
Actual: YES	FN = 15 200 (23%)	TP = 23 747 (36%)	38 947 (59%)
	32 594 (49%)	33 594 (51%)	

Table 4.44: confusion-matrix - computational AT irrecoverable + defaulting - B) - version i

To be able to do the comparison, let's calculate only the right side again as a total of 100%, the result is as per following matrix 4.45:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 9 847 (29% +)	N/A
Actual: YES	FN = N/A	TP = 23 747 (71% +)	N/A
	N/A	33 594 (100% +)	

Table 4.45: confusion-matrix - computational AT irrecoverable + defaulting - B) version ii

C) NULL is low risk - Scenario I - confusion-matrix - for cases when only irrecoverable is the bad case 4.46 :

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 3 119 (5%)	FP = 2 764 (4%)	5 883 (9%)
Actual: YES	FN = 22 366 (34%)	TP = 37 939 (57%)	60 305 (91%)
	25 485 (39%)	40 703 (61%)	

Table 4.46: confusion-matrix - computational AT irrecoverable - C) - version i

To be able to do the comparison, let's calculate only the right side as a total of 100%, this can be seen in the following matrix 4.47:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 2 764 (7% +)	N/A
Actual: YES	FN = N/A	TP = 37 939 (93% +)	N/A
	N/A	40 703 (100% +)	

Table 4.47: confusion-matrix - computational AT irrecoverable - C) - version ii

C) NULL is low risk - Scenario II - confusion-matrix - for cases when irrecoverable and defaulting together are the bad cases 4.48:

n = 66 188	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 10 285 (16%)	FP = 16 956 (26%)	27 241 (41%)
Actual: YES	FN = 15 200 (23%)	TP = 23 747 (36%)	38 947 (59%)
	25 485 (39%)	40 703 (61%)	

Table 4.48: confusion-matrix - computational AT irrecoverable + defaulting - C) - version i

To be able to do the comparison, let's calculate only the right side again as a total of 100%, the result is as per following matrix 4.49:

n = N/A	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = N/A	FP = 16 956 (42% +)	N/A
Actual: YES	FN = N/A	TP = 23 747 (58% +)	N/A
	N/A	40 703 (100% +)	

Table 4.49: confusion-matrix - computational AT irrecoverable + defaulting - C) version ii

4.7 Layer 7. - Comparison

The comparison of grounded and preferred semantics as per following table 4.50 :

Returns results	**Grounded semantics**	**Preferred semantics**
Amount of cases	66 188	59 079
Percentage of cases	100%	87%

Table 4.50: Comparison - Grounded vs. Preferred semantics

(a) Grounded semantics:

The comparison between the Scorecard method and Computational Argumentation Theory - Grounded semantics can be seen in the summary table below 4.51 . The winner is always that method, where the + number is higher :

Feature	**Scorecard**	**Computational AT**	**Comment**
Scenario I: Irrecoverable is the bad case	+: 90% ; -: 10%	+: 91% ; -: 9%	AT wins
Scenario II: Irrecoverable + default are the bad cases	+: 66% ; -: 34%	+: 60% ; -: 40%	Scorecard wins

Table 4.51: Comparison - Scorecard vs. Computational AT (Grounded semantics)

(b) Preferred semantics:

The comparison between the Scorecard method and Computational Argumentation Theory - Grounded semantics can be seen in the summary table below 4.52 . The winner is always that method, where the + number is higher :

Case	**Feature**	**Scorecard**	**Computational AT**	**Comment**
A)	**Scenario I: Irrec is the bad case**	+: 90% ; -: 10%	+: 92% ; -: 8%	AT wins
A)	**Scenario II: Irrec + default are the bad cases**	+: 66% ; -: 34%	+: 71% ; -: 29%	AT wins
B)	**Scenario I: Irrec is the bad case**	+: 90% ; -: 10%	+: 91% ; -: 9%	AT wins
B)	**Scenario II: Irrec + default are the bad cases**	+: 66% ; -: 34%	+: 71% ; -: 29%	AT wins
C)	**Scenario I: Irrec is the bad case**	+: 90% ; -: 10%	+: 93% ; -: 7%	AT wins
C)	**Scenario II: Irrec + default are the bad cases**	+: 66% ; -: 34%	+: 58% ; -: 42%	Scorecard wins

Table 4.52: Comparison - Scorecard vs. Computational AT (Preferred semantics

Chapter 5

Analysis, evaluation and discussion

5.1 Hypothesis testing

The H0 (H null) hypothesis is: *The inferences produced by a model built using computational AT (Argumentation Theory) are expected to over perform the Scorecard method, in terms of default loan prediction.*

The visual representation of the result summary tables (refer to previous chapter: 4.51 and 4.52) can be seen on the following graph 5.1:

The results summary graph below (5.1) displays 8 different scenarios which were analysed and in 6 cases out of 8 cases (75%) computational Argumentation Theory is the winner and in 2 cases out of 8 cases (25%) the Scorecard Method is the winner. If we consider that the majority wins, then AT is the overall winner, and therefore we can accept Hypothesis null and we can reject Hypothesis one.

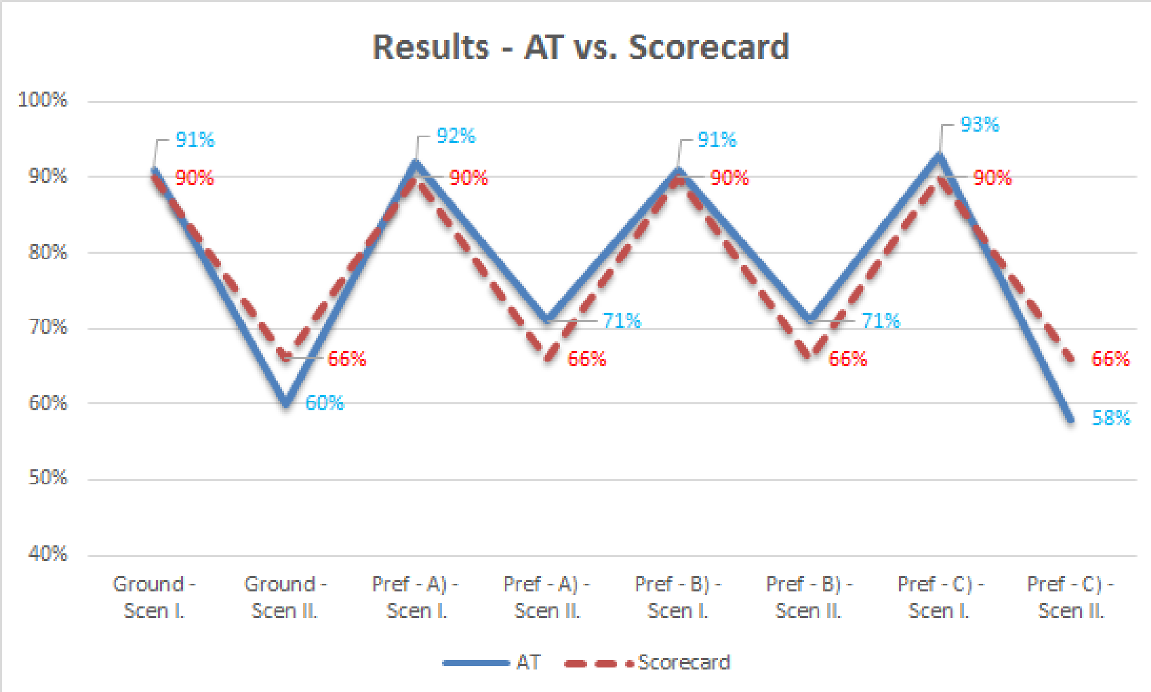

Figure 5.1: Result summary - AT vs. Scorecard

5.2 Link research question

The research question is *Can the use of Defeasible Reasoning and Computational Argumentation Theory increase representation and assessment the risk of mortgages, compared to the scorecard method?*

Looking at the experiment executed, the results of the experiment and the outcoming evidence provided, the answer is:

Yes, Defeasible Reasoning and Computational Argumentation Theory can successfully represent and measure risk assessment for mortgages, in Ireland.

5.3 Benchmark - Scorecard method

The scorecard method predicted better, against Grounded semantics when irrecoverable and defaulting were considered as the bad cases; and also predicted better against Preferred semantics, in the same scenario , when defaulting and irrecoverable were considered as bad cases together, but only when the missing values for Preferred semantics were considered as low risk cases. Scorecard method is still the bank industry standard, it is heavily regulated by the financial regulator authorities which makes it comparable across banks, but it also has the limitations, in case a better process becomes available, it might take time to be officially accepted, considering enough evidence was provided.

5.4 Grounded semantics

The grounded semantics is a more comprehensive semantics, than the preferred semantics, even though it represents a sceptical view, as it returns a result for every case. By looking at strict business view, which is only the irrecoverable cases are the bad cases, then it performs well. However, when the irrecoverable and defaulting cases are considered together to be the bad cases, then the scorecard method is outperforming the grounded semantics.

5.5 Preferred semantics

The preferred semantics represents the credulous view and it is true for the majority of the cases, as in 5 out of 6 cases it over performs the scorecard method. It has to be noted, that using the preferred semantics, for 13% of the cases there will be no result presented. That is due to cases, which can not be resolved with preferred semantics, as the premises can not defend themselves against some other arguments and also those arguments can not defend themselves against the premises. Regardless of how are these missing values treated, the result which is returned is still a good result, except when the missing cases are treated as low risk cases and irrecoverable and defaulting are considered together to be the bas cases.

5.6 Strengths and limitations of findings

The strength of the study is about the representation and visualisation of DR and the preparation of a computational AT approach for assessing the mortgage applications, which is a new aspect of risk analysis and a new way of predicting the risk of defaulting. The proof of concept works, it provides the results which were expected, even though in some cases actually the scorecard method predicts better. The study was limited to the True Positives and the False Positives in the confusion-matrix, because those cases where the applicants were not awarded with the mortgage, those cases were not part of the data set, which was received from the medium size Irish bank. The outcome depends on which semantics were chosen for the calculation, and it also depends on the definition which cases are considered to be defaulting: the irrecoverable only, or the irrecoverable and defaulting together. For those cases where the model does not return a result, the outcome could change depending on how are those missing values treated.

The biggest limitation is, that the data set only contains live mortgage applications. Hence the study can only deal with cases which were already accepted by the bank and the applicants were awarded with the mortgage loan. There is no information in the data set about the cases where the mortgage was declined. Another big limitation is, that in 41% of the cases the Purchase Price of the property is missing. As this would be a key feature to calculate the ratio of the loan compared to the PDH (Primary Dwelling Home) this is hugely influencing the outcome of the prediction. It is believed, that a lower credit percentage would represent a lower risk, but if this information is missing in nearly every second case, that limits the accuracy of the prediction. As we can see in the result summary tables (4.51 and 4.52) the outcome also depends on the semantics selected, and whether we consider the bad cases as irrecoverable only, or the irrecoverable and defaulting together. Based on the combination of these, the winner is different. The sample size of over 60k records is sufficient enough to prepare a model. The more data is available, the more model results can be obtained. The computational time for calculating the results using a web ui tool takes a few hours for the 60k+ records, and because of the constant timeout error messages the process had to be divided to sub-processes and the gathered results aggregated. In case there was a mistake everything had to be re-calculated again, which caused extra few hours processing time every time this happened. It could be advantageous if the processing times could be accelerated. Beside hardware and software improvement, there would be another way to speed up the calculations, which is reviewing the argument graph to see if it would be possible to reduce the amount of attacks and simplify the arguments, but without causing any loss to the functionality. If that would be possible, that should speed up the calculations radically.

The computational AT approach can represent every aspect and every stage of the mortgage application, using defeasible reasoning, because of the similarities to real life, i.e. when new evidence becomes available, that could change the conclusion.

Chapter 6

Conclusion

6.1 Research Overview

Based on reviewing the available journals and articles; the internal bank documentation; and the data set which was provided by the medium size Irish bank, 7 layers were established on how to implement DR in practice with computational AT approach. These 7 layers are: 1) Data understanding, 2) Data preparation, 3) Internal structure of arguments, 4) Conflicts of arguments, 5) Dialectical status of arguments, 6) Accrual of the acceptable arguments, and 7) Comparison. Each of these layers were first designed and then implemented.

6.2 Problem Definition

The research problem what this paper intended to resolve is, whether it is possible to create a computational AT approach for implementing DR to predict the PD of mortgage applications in Ireland. This is currently not in the literature. The current PD prediction method, the Scorecard has several limitations because of using a strict formula one fits all and there is very little space for exception handling. Beside the internal limitations of the Scorecard method itself, there is also a market requirement which would desire a more scientific approach with a high accuracy of risk prediction.

6.3 Design/Experimentation, Evaluation & Results

Based on the evidence of the experiment, it can be determined, that DR and computational AT can be used successfully to model the predicting of risk of defaulting, and in 6 out of 8 cases (that is 75%), where AT predicts better than the regular scorecard method. Depending on which semantics is used for the final calculation of the accrual of the acceptable arguments, the model's PD score is changing. For this study there are 2 relevant semantics the Grounded semantics and the Preferred semantics. The grounded semantics predicts better in a strict business case, when only the irrecoverable cases are considered as bad result. The preferred semantics returns no result in 13% of the cases, due to the lack of resolution using it's algorithm, however, depending on how these missing values are treated, except in the case when missing values are considered as low risk and also irrecoverable and defaulting cases are considered together to be the business problems, in the other 5 cases the preferred semantics predicted better than the scorecard method.

6.4 Contributions and impact

The study is providing a new way of assessing the risk of mortgage applications, on a way, which has never been done before, using state-of-the-art methods with Defeasible Reasoning and computational Argumentation Theory. While the bank's Scorecard method is limited to be only a number, as a result of a formula, AT and DR are much more than that. The entire mortgage lending process could be visualised and implemented with AT and DR, ready for several alternative routes as it is possible to create more graphs in the AT approach. The term *when new evidence becomes available*, the entire data and attachment collection process can be symbolised with AT, as it is a new evidence every single time. Even more, on an interactive way, once this new piece of evidens is added, the graph could pick it up and could recalculate the premises and the conclusion immediately. Currently there is no one page summary as an overview for a mortgage application at the bank and this could be the solution.

6.5 Future Work & recommendations

It might be worth to have and compare different graphs, based on the interpretation of the rules and the problem. To have more graphs would give a stronger evidence, whether AT is good or is not for the purpose, as only one might not be enough to confirm.

It would be also worth to create more cycles to improve the rules, and to improve the labeling logic condition details (in, out, undec). To improve the accruals calculation, other mathematical methods, like the weighted average or median also could be considered, to see if the model predicts better with those new enhanced features.

To increase the evaluating scenarios and to present a more colourful and realistic picture in predicting, more used cases could be presented, based on the general level of risk taking, i.e. a graph could be prepared which represents a more dangerous so more risky situation and another one which visualizes a more strict and more secure method.

Attigeri, Manohara Pai M M, Pai, and Nayak (2015) conducted a big data approach research on another area of the financial market, the stock exchange market, and the finding is, that combining available historical data with news feed and social media data, there is a significant correlation and the accuracy of the prediction is much higher than simply relying on the internal system's database data. A similar approach could be conducted as a future research using Defeasible Reasoning and building a model for computational Argumentation Theory, with external social media, news, competitor analysis, market analysis and cross-country data.

References

Antoniou, G., & Bikakis, A. (2007). DR-prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. , *19*(2), 233–245. Retrieved 2016-11-27, from http://ieeexplore.ieee.org/document/4039286/ doi: 10.1109/TKDE.2007.29

Attigeri, G. V., Manohara Pai M M, Pai, R. M., & Nayak, A. (2015). Stock market prediction: A big data approach. In (pp. 1–5). IEEE. Retrieved 2016-11-12, from http://ieeexplore.ieee.org/document/7373006/ doi: 10.1109/TENCON.2015.7373006

Bellucci, E., Lodder, A., & Zeleznikow, J. (2004). Integrating artificial intelligence, argumentation and game theory to develop an online dispute resolution environment. In (pp. 749–754). IEEE Comput. Soc. Retrieved 2016-11-12, from http://ieeexplore.ieee.org/document/1374264/ doi: 10.1109/ICTAI.2004.75

Berariu, T. (2014). An argumentation framework for BDI agents. In F. Zavoral, J. J. Jung, & C. Badica (Eds.), *Intelligent distributed computing VII* (Vol. 511, pp. 343–354). Springer International Publishing. Retrieved 2017-07-08, from http://link.springer.com/10.1007/978-3-319-01571-2_40 (DOI: 10.1007/978-3-319-01571-2_40)

Bikakis, A., & Antoniou, G. (2010). Defeasible contextual reasoning with arguments in ambient intelligence. , *22*(11), 1492–1506. Retrieved 2016-11-27, from http://ieeexplore.ieee.org/document/5416723/ doi: 10.1109/TKDE.2010.37

Bikakis, A., & Antoniou, G. (2011). Contextual defeasible logic and its application to ambient intelligence. , *41*(4), 705–716. Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/document/5756697/` doi: 10.1109/TSMCA.2011.2132715

Bilo Doumbouya, M., Kamsu-Foguem, B., & Kenfack, H. (2016). Argumentation and graph properties. , *52*(2), 319–325. Retrieved 2016-11-12, from `http://linkinghub.elsevier.com/retrieve/pii/S0306457315001041` doi: 10.1016/j.ipm.2015.08.003

Bisquert, P., Cayrol, C., Saint-Cyr, F. D. D., & Lagasquie, M.-C. (2013). Goal-driven changes in argumentation: A theoretical framework and a tool. In (pp. 610–617). IEEE. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6735307/` doi: 10.1109/ICTAI.2013.96

Brasil Jr, S. M. (2001). Rules and principles in legal reasoning. a study of vagueness and collisions in artificial intelligence and law. , *Vol. 10.*, p67–77. 11p.

Carenini, G., & Moore, J. D. (2006). Generating and evaluating evaluative arguments. , *170*(11), 925–952. Retrieved 2016-11-12, from `http://linkinghub.elsevier.com/retrieve/pii/S000437020600066X` doi: 10.1016/j.artint.2006.05.003

Chen, Y., Chen, C., & Yao, X. (2010). Analysis of the causes and effects of the global financial crisis and lessons learned. In (pp. 1–4). IEEE. Retrieved 2017-07-07, from `http://ieeexplore.ieee.org/document/5576529/` doi: 10.1109/ICMSS.2010.5576529

Chesnevar, C., & Maguitman, A. (2004). ArgueNet: an argument-based recommender system for solving web search queries. In (pp. 282–287). IEEE. Retrieved 2016-12-18, from `http://ieeexplore.ieee.org/document/1344683/` doi: 10.1109/IS.2004.1344683

Cristani, M., Tomazzoli, C., Karafili, E., & Olivieri, F. (2016). Defeasible reasoning about electric consumptions. In (pp. 885–892). IEEE. Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/document/7474183/` doi: 10.1109/AINA.2016.62

Dauphin, J., & Schulz, C. (2014). Arg teach - a learning tool for argumentation theory. In (pp. 776–783). IEEE. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6984556/` doi: 10.1109/ICTAI.2014.120

Davy, D. (2011). Canada's housing bubble: A rhetorical analysis. In (pp. 1–6). IEEE. Retrieved 2017-07-07, from `http://ieeexplore.ieee.org/document/6087200/` doi: 10.1109/IPCC.2011.6087200

Devasena, C. L. (2014). Adeptness evaluation of memory based classifiers for credit risk analysis. In (pp. 143–147). IEEE. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6965029/` doi: 10.1109/ICICA.2014.39

Dukic, D., Dukic, G., & Kvesic, L. (2011). A credit scoring decision support system. In *Proceedings of the ITI 2011* (pp. Pages: 391 – 396). Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5974054`

Dung, P. M. (1995a). An argumentation-theoretic foundation for logic programming. , *22*(2), 151–177. Retrieved 2017-07-31, from `http://linkinghub.elsevier.com/retrieve/pii/074310669594697X` doi: 10.1016/0743-1066(95)94697-X

Dung, P. M. (1995b). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. , *77*(2), 321–357. Retrieved 2017-07-31, from `http://linkinghub.elsevier.com/retrieve/pii/000437029400041X` doi: 10.1016/0004-3702(94)00041-X

Eemeren, F. H. v., Grootendorst, R., Johnson, R. H., Plantin, C., & Willard, C. A. (2013). *Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments*. Routledge.

Fan, M., Yang, S., & Zhang, J. (2008). Credit rating models considering sample selection biases. In (pp. 433–436). IEEE. Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/document/5116512/` doi: 10.1109/ISBIM.2008.235

Janjua, N. K., & Hussain, F. K. (2011). Defeasible reasoning based argumentative web-IDSS for virtual teams (VTs). In (pp. 330–334). IEEE. Retrieved 2016-11-

27, from `http://ieeexplore.ieee.org/document/6040872/` doi: 10.1109/WI-IAT.2011.168

Kravari, K., Kontopoulos, E., & Bassiliades, N. (2009). Towards a knowledge-based framework for agents interacting in the semantic web. In (pp. 482–485). IEEE. Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/document/5285132/` doi: 10.1109/WI-IAT.2009.198

Longo, L. (2012). Formalising human mental workload as non-monotonic concept for adaptive and personalised web-design. In J. Masthoff, B. Mobasher, M. C. Desmarais, & R. Nkambou (Eds.), *User modeling, adaptation, and personalization* (Vol. 7379, pp. 369–373). Springer Berlin Heidelberg. Retrieved 2017-07-30, from `http://link.springer.com/10.1007/978-3-642-31454-4_38` (DOI: 10.1007/978-3-642-31454-4_38)

Longo, L. (2014). *Formalising human mental workload as a defeasible computational concept* (Doctoral Thesis). Retrieved 2017-07-30, from `http://arrow.dit.ie/scschcomoth/10/`

Longo, L. (2015). A defeasible reasoning framework for human mental workload representation and assessment. , *34* (8), 758–786. Retrieved 2016-11-12, from `http://www.tandfonline.com/doi/full/10.1080/0144929X.2015.1015166` doi: 10.1080/0144929X.2015.1015166

Longo, L. (2016a). Argumentation for knowledge representation, conflict resolution, defeasible inference and its integration with machine learning. In A. Holzinger (Ed.), *Machine learning for health informatics* (Vol. 9605, pp. 183–208). Springer International Publishing. Retrieved 2017-07-08, from `http://link.springer.com/10.1007/978-3-319-50478-0_9` (DOI: 10.1007/978-3-319-50478-0_9)

Longo, L. (2016b). Mental workload in medicine: Foundations, applications, open problems, challenges and future perspectives. In (pp. 106–111). IEEE. Retrieved 2017-07-17, from `http://ieeexplore.ieee.org/document/7545966/` doi: 10.1109/CBMS.2016.36

Longo, L., & Dondio, P. (2014). Defeasible reasoning and argument-based systems in medical fields: An informal overview. In (pp. 376–381). IEEE. Retrieved 2016-12-18, from http://ieeexplore.ieee.org/document/6881909/ doi: 10.1109/CBMS.2014.126

Longo, L., & Hederman, L. (2013). Argumentation theory for decision support in health-care: A comparison with machine learning. In K. Imamura, S. Usui, T. Shirao, T. Kasamatsu, L. Schwabe, & N. Zhong (Eds.), *Brain and health informatics* (Vol. 8211, pp. 168–180). Springer International Publishing. Retrieved 2017-07-30, from http://link.springer.com/10.1007/978-3-319-02753-1_17 (DOI: 10.1007/978-3-319-02753-1_17)

Longo, L., Kane, B., & Hederman, L. (2012). Argumentation theory in health care. In (pp. 1–6). IEEE. Retrieved 2017-07-30, from http://ieeexplore.ieee.org/document/6266323/ doi: 10.1109/CBMS.2012.6266323

Murphy, E., & Scott, M. (2013). Mortgage-related issues in a crisis economy: Evidence from rural households in ireland. , *46*, 34–44. Retrieved 2017-07-04, from http://linkinghub.elsevier.com/retrieve/pii/S0016718512002771 doi: 10.1016/j.geoforum.2012.12.003

Nute, D., Henderson, C., & Hunter, Z. (1997). d-graph: an argument-based system incorporating defeasible graphs. In (Vol. 5, pp. 354–362). IEEE Comput. Soc. Press. Retrieved 2016-12-18, from http://ieeexplore.ieee.org/document/663193/ doi: 10.1109/HICSS.1997.663193

Panisson, A. R., & Bordini, R. H. (2016). Knowledge representation for argumentation in agent-oriented programming languages. In (pp. 13–18). IEEE. Retrieved 2017-07-08, from http://ieeexplore.ieee.org/document/7839555/ doi: 10.1109/BRACIS.2016.014

Quan, T. T., Luong, X. H., Nguyen, T. C., & Cheung, H. S. (2015). Argumentation-based schema matching for multiple digital libraries. , *39*(1), 81–103. Retrieved 2016-11-12, from http://www.emeraldinsight.com/doi/10.1108/OIR-02-2014-0023 doi: 10.1108/OIR-02-2014-0023

Rakib, A., & Haque, H. M. U. (2015). Modeling and verifying context-aware non-monotonic reasoning agents. In (pp. 61–69). IEEE. Retrieved 2016-11-27, from http://ieeexplore.ieee.org/document/7340471/ doi: 10.1109/MEMCOD.2015.7340471

Rizzo, L., Dondio, P., Delany, S. J., & Longo, L. (2016). Modeling mental workload via rule-based expert system: A comparison with NASA-TLX and workload profile. In L. Iliadis & I. Maglogiannis (Eds.), *Artificial intelligence applications and innovations* (Vol. 475, pp. 215–229). Springer International Publishing. Retrieved 2017-07-30, from http://link.springer.com/10.1007/978-3-319-44944-9_19 (DOI: 10.1007/978-3-319-44944-9_19)

Rizzo, L., & Longo, L. (2017). Representing and inferring mental workload via defeasible reasoning: A comparison with the NASA task load index and the workload profile. In *Proceedings of the 1st workshop on advances in argumentation in artificial intelligence co-located with XVI international conference of the italian association for artificial intelligence (ai*ia 2017), bari, italy, november 16-17, 2017.* (pp. 126–140). Retrieved from http://ceur-ws.org/Vol-2012/AI3-2017_paper_13.pdf

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiersa survey. , *35*(4), 476–487. Retrieved 2016-11-12, from http://ieeexplore.ieee.org/document/1522531/ doi: 10.1109/TSMCC.2004.843247

Tomic, T. (2010). The philosophy of information as an underlying and unifying theory of information science. , *Vol. 15.*, p2–2. 1p. Retrieved 2016-11-12, from http://www.informationr.net/ir/15-4/colis714.html

Turkson, R. E., Baagyere, E. Y., & Wenya, G. E. (2016). A machine learning approach for predicting bank credit worthiness. In (pp. 1–7). IEEE. Retrieved 2016-11-12, from http://ieeexplore.ieee.org/document/7585216/ doi: 10.1109/ICAIPR.2016.7585216

Vagin, V., & Morosin, O. (2013). Modeling defeasible reasoning for argumentation. In (pp. 304–309). IEEE. Retrieved 2016-11-27, from `http://ieeexplore.ieee.org/document/6855867/` doi: 10.1109/BRICS-CCI-CBIC.2013.58

Waldron, R., & Redmond, D. (2014). The extent of the mortgage crisis in ireland and policy responses. , *29*(1), 149–165. Retrieved 2017-07-04, from `http://www.tandfonline.com/doi/abs/10.1080/02673037.2013.825694` doi: 10.1080/02673037.2013.825694

Wang, X., Jeong, D., Chang, R., & Ribarsky, W. (2012). RiskVA: A visual analytics system for consumer credit risk analysis. , *17*(4), 440–451. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6297590/` doi: 10.1109/TST.2012.6297590

Yaohua, S. (2012). Research on the theoretical model of psychological factors of borrowers with residential mortgage loan. In (pp. 175–178). IEEE. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6382493/` doi: 10.1109/BCGIN.2012.52

Youqin Pan. (2016). Sustainability trends in financial services sector: Evidence from europe and north america. In (pp. 1–6). IEEE. Retrieved 2017-07-07, from `http://ieeexplore.ieee.org/document/7538549/` doi: 10.1109/ICSSSM.2016.7538549

Zeleznikow, J., & Stranieri, A. (1998). Split up: an intelligent decision support system which provides advice upon property division following divorce. , *Vol. 6.*, p190. 24p.

Zha, K.-L. (2016). Modeling and analysis of multi-investment risk evaluation based on artificial neural network. In (pp. 255–259). IEEE. Retrieved 2017-07-30, from `http://ieeexplore.ieee.org/document/7757121/` doi: 10.1109/ICRIS.2016.51

Zheng-xuan, Z., & Zu, Y. (2013). Prediction model of mortgage prepayment risk. In (pp. 418–424). IEEE. Retrieved 2016-11-12, from `http://ieeexplore.ieee.org/document/6586315/` doi: 10.1109/ICMSE.2013.6586315

Zhu, X., & Wang, P. (2008). The credit risk assessment method based on evidence reasoning. In (pp. 511–514). IEEE. Retrieved 2017-07-08, from http://ieeexplore.ieee.org/document/4731989/ doi: 10.1109/IITA.Workshops.2008.34

Appendix A

Additional content - definition of arguments

A.1 SQL code - Argument 01

```sql
SELECT
    [AGREEMENT_NO]
    ,[PRIMARY_OCCUPATION]
    ,[primoccup_group]
    ,[grp_primoccup]
    ,[primoccup_group] AS AT_RiskLevel
    ,CASE
        WHEN [primoccup_group] = 1 THEN 'Low Risk'
        WHEN [primoccup_group] = 2 THEN 'Medium-Low Risk'
        WHEN [primoccup_group] = 3 THEN 'Medium Risk'
        WHEN [primoccup_group] = 4 THEN 'Medium-High Risk'
        WHEN [primoccup_group] >= 5 THEN 'High Risk'
    END AS RiskCateg
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.1: SQL - Argument 01

A.2 SQL code - Argument 02

```sql
SELECT
    [AGREEMENT_NO]
    ,[REGION_NAME]
    ,[region_group]
    ,[new_grp_region]

    ,CASE
        WHEN [REGION_NAME] like '%DUBLIN%' THEN 1
        WHEN [REGION_NAME] IN
        (
        'CO LOUTH' ,'CO MEATH' ,'CO KILDARE' ,'CO WICKLOW' ,'CO CORK' ,'CO GALWAY' ,'CORK 14' ,'CORK CITY' ,'GALWAY CITY' ,
        'LIMERICK CITY' ,'WATERFORD CITY' ,'ADARE' ,'ASHBOURNE' ,'CARLOW TOWN' ,'CELBRIDGE' ,'CORK 1' ,'CORK CITY' ,
        'DROGHEDA' ,'GALWAY CITY' ,'KILKENNY CITY' ,'LIMERICK 1' ,'LIMERICK CITY' ,'MAYNOOTH' ,'NAAS' ,'NAVAN' ,'SWORDS' ,
        'WATERFORD CITY' ,'WICKLOW TOWN'
        )
        THEN 2
        WHEN ([REGION_NAME] not like '%DUBLIN%') AND ([REGION_NAME] NOT IN
        (
        'CO LOUTH' ,'CO MEATH' ,'CO KILDARE' ,'CO WICKLOW' ,'CO CORK' ,'CO GALWAY' ,'CORK 14' ,'CORK CITY' ,'GALWAY CITY' ,
        'LIMERICK CITY' ,'WATERFORD CITY' ,'ADARE' ,'ASHBOURNE' ,'CARLOW TOWN' ,'CELBRIDGE' ,'CORK 1' ,'CORK CITY' ,
        'DROGHEDA' ,'GALWAY CITY' ,'KILKENNY CITY' ,'LIMERICK 1' ,'LIMERICK CITY' ,'MAYNOOTH' ,'NAAS' ,'NAVAN' ,'SWORDS' ,
        'WATERFORD CITY' ,'WICKLOW TOWN'
        )
        ) THEN 3
        ELSE [region_group]
    END AS AT_RiskLevel

FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.2: SQL - Argument 02

A.3 SQL code - Argument 03

```sql
SELECT
    [AGREEMENT_NO]
    ,[Joint_Total_Income]
    ,[AMOUNT_DRAWN]

    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) < (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) THEN 1
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) THEN 2
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) THEN 3
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0) THEN 4
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) > (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0) THEN 5
    END AS AT_RiskLevel

    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) < (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) THEN 'Low Risk'
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) THEN 'Medium-Low Risk'
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) THEN 'Medium Risk'
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0) THEN 'Medium-High Risk'
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) > (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0) THEN 'High Risk'
    END AS RiskCateg

FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.3: SQL - Argument 03

A.4 SQL code - Argument 04

```sql
SELECT
    [AGREEMENT_NO]
    ,[LOAN_TERM]

    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) <= 35 THEN 1
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) > 35 THEN 5
    END AS AT_RiskLevel

    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) <= 35 THEN 'Low Risk'
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) > 35 THEN 'High Risk'
    END AS RiskCateg
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.4: SQL - Argument 04

A.5 SQL code - Argument 05

```sql
SELECT
    [AGREEMENT_NO]
    ,[PRIMARY_DATE_OF_BIRTH]
    ,[COMPLETION_DATE]
    ,[LOAN_TERM]
    ,AgeAtCompletion
    ,Age_Plus_LoanTerm

    ,CASE WHEN Age_Plus_LoanTerm < 20 THEN 5
        WHEN Age_Plus_LoanTerm >= 20 AND Age_Plus_LoanTerm <= 40 THEN 1
        WHEN Age_Plus_LoanTerm > 40 AND Age_Plus_LoanTerm <= 50 THEN 2
        WHEN Age_Plus_LoanTerm > 50 AND Age_Plus_LoanTerm <= 60 THEN 3
        WHEN Age_Plus_LoanTerm > 60 AND Age_Plus_LoanTerm <= 70 THEN 4
        WHEN Age_Plus_LoanTerm > 70 THEN 5
    END AS AT_RiskLevel

FROM
(
SELECT
    [AGREEMENT_NO]
    ,[PRIMARY_DATE_OF_BIRTH]
    ,[COMPLETION_DATE]
    ,[LOAN_TERM]
    ,(DATEDIFF(YEAR, CONVERT(DATE,[COMPLETION_DATE]) , (CONVERT(DATE,[PRIMARY_DATE_OF_BIRTH])) ) *-1 ) AS AgeAtCompletion
    ,((DATEDIFF(YEAR, CONVERT(DATE,[COMPLETION_DATE]) , (CONVERT(DATE,[PRIMARY_DATE_OF_BIRTH])) ) *-1 )
     + CONVERT(NUMERIC(10,2),[LOAN_TERM])) AS Age_Plus_LoanTerm

FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) Table1
```

Figure A.5: SQL - Argument 05

A.6 SQL code - Argument 06

```sql
SELECT
    *
    ,CASE
        WHEN RepaymentYearly *2 < NDI_Yearly THEN 1
        WHEN RepaymentYearly *2 > NDI_Yearly AND NDI_Yearly >= RepaymentYearly *1.6 THEN 2
        WHEN RepaymentYearly *1.6 > NDI_Yearly AND NDI_Yearly >= RepaymentYearly *1.2 THEN 3
        WHEN RepaymentYearly *1.2 > NDI_Yearly AND NDI_Yearly >= RepaymentYearly *0.60 THEN 4
        WHEN RepaymentYearly *0.60 > NDI_Yearly THEN 5
        WHEN (NDI_Yearly = 0) THEN 5
    END AT_RiskLevel
FROM
(
SELECT
    [AGREEMENT_NO]
    ,[Joint_Total_Income]
    ,[AMOUNT_DRAWN]
    ,[LOAN_TERM]

    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 30000 THEN 0
        WHEN 30000 <= CONVERT(NUMERIC(10,2),[Joint_Total_Income])
        AND  CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 40000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.3),2)
        WHEN 40000 <= CONVERT(NUMERIC(10,2),[Joint_Total_Income])
        AND  CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 60000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.4),2)
        WHEN CONVERT(NUMERIC(10,2),[Joint_Total_Income]) >= 60000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.5),2)
    END AS NDI_Yearly
    ,ROUND(((CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))*(POWER(1.02,CONVERT(NUMERIC(10,2),[LOAN_TERM])))),2) AS PaybackAmount
    ,ROUND((ROUND(((CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))*(POWER(1.02,CONVERT(NUMERIC(10,2),[LOAN_TERM])))),2)
     / CONVERT(NUMERIC(10,2),[LOAN_TERM])),2) AS RepaymentYearly
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) TableA
```

Figure A.6: SQL - Argument 06

A.7 SQL code - Argument 07

```
SELECT
    [AGREEMENT_NO] ,[LOAN_PURPOSE] ,FirstTimeBuyerFlag ,[AMOUNT_DRAWN] ,[PURCHASE_PRICE]
    ,[AMOUNT_DRAWN_NUM] ,[PURCHASE_PRICE_NUM] ,[PP_DivBy_AD],
    CASE WHEN [Joint_Total_Income] > 47000 THEN
    CASE        WHEN [PP_DivBy_AD] <= 1.1 THEN 5
                WHEN [PP_DivBy_AD] > 1.1 AND [PP_DivBy_AD] <= 1.2 THEN 4
                WHEN [PP_DivBy_AD] > 1.2 AND [PP_DivBy_AD] <= 1.3 THEN 3
                WHEN [PP_DivBy_AD] > 1.3 AND [PP_DivBy_AD] <= 1.40 THEN 2
                WHEN [PP_DivBy_AD] > 1.40 THEN 1   END
    ELSE CASE   WHEN [PP_DivBy_AD] <= 1.4 THEN 5
                WHEN [PP_DivBy_AD] > 1.4 AND [PP_DivBy_AD] <= 1.5 THEN 4
                WHEN [PP_DivBy_AD] > 1.5 AND [PP_DivBy_AD] <= 1.6 THEN 3
                WHEN [PP_DivBy_AD] > 1.6 AND [PP_DivBy_AD] <= 1.7 THEN 2
                WHEN [PP_DivBy_AD] > 1.7 THEN 1   END
    END AS AT_RiskLevel
FROM
(
SELECT
    [AGREEMENT_NO] ,[LOAN_PURPOSE]
    ,CASE WHEN [LOAN_PURPOSE] LIKE '%First Time Buyer%' THEN 1 ELSE 0 END AS FirstTimeBuyerFlag
    ,[AMOUNT_DRAWN] ,[PURCHASE_PRICE]
    ,CONVERT(NUMERIC(10,2),[Joint_Total_Income]) AS [Joint_Total_Income]
    ,CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) AS [AMOUNT_DRAWN_NUM]
    ,CONVERT(NUMERIC(10,2),[PURCHASE_PRICE]) AS [PURCHASE_PRICE_NUM]
    ,CONVERT(NUMERIC(10,2),[PURCHASE_PRICE]) / (CASE WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])=0 THEN 1
     ELSE CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) END) AS [PP_DivBy_AD]
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) Table1
```

Figure A.7: SQL - Argument 07

A.8 SQL code - Argument 08

```
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN [primoccup_group] = 1 THEN 1
        WHEN [primoccup_group] = 2 THEN 2
        WHEN [primoccup_group] = 3 THEN 3
        WHEN [primoccup_group] = 4 THEN 4
        WHEN [primoccup_group] = 5 AND
         [PRIMARY_SELF_EMPLOYED] = 'E' AND
         CONVERT(NUMERIC(10,2),[Joint_Total_Income]) >= 50000 THEN 4
        WHEN [primoccup_group] = 5 AND
         ([PRIMARY_SELF_EMPLOYED] != 'E' OR
         ISNULL(CONVERT(NUMERIC(10,2),[Joint_Total_Income]),0) < 50000) THEN 5
    END AS AT_RiskLevelNew
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.8: SQL - Argument 08

A.9 SQL code - Argument 09

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE WHEN ISNULL(CONVERT(NUMERIC(20,0),[PURCHASE_PRICE]),0) = 0 THEN 5
        ELSE
            CASE WHEN (((CONVERT(NUMERIC(20,2),[AMOUNT_DRAWN])) / (CONVERT(NUMERIC(20,2),ISNULL([PURCHASE_PRICE],0))))) < 0.7 THEN
                CASE WHEN [REGION_NAME] like '%DUBLIN%' THEN 1
                    WHEN [REGION_NAME] not like '%DUBLIN%' THEN 2
                END
            ELSE

            CASE
        WHEN [REGION_NAME] like '%DUBLIN%' THEN 1
        WHEN [REGION_NAME] IN
        ('CO LOUTH' ,'CO MEATH' ,'CO KILDARE' ,'CO WICKLOW' ,'CO CORK' ,'CO GALWAY' ,'CORK 14' ,'CORK CITY' ,'GALWAY CITY' ,
        'LIMERICK CITY' ,'WATERFORD CITY' ,'ADARE' ,'ASHBOURNE' ,'CARLOW TOWN' ,'CELBRIDGE' ,'CORK 1' ,'CORK CITY' ,
        'DROGHEDA' ,'GALWAY CITY' ,'KILKENNY CITY' ,'LIMERICK 1' ,'LIMERICK CITY' ,'MAYNOOTH' ,'NAAS' ,'NAVAN' ,'SWORDS' ,
        'WATERFORD CITY' ,'WICKLOW TOWN')   THEN 2
        WHEN ([REGION_NAME] not like '%DUBLIN%') AND ([REGION_NAME] NOT IN
        ('CO LOUTH' ,'CO MEATH' ,'CO KILDARE' ,'CO WICKLOW' ,'CO CORK' ,'CO GALWAY' ,'CORK 14' ,'CORK CITY' ,'GALWAY CITY' ,
        'LIMERICK CITY' ,'WATERFORD CITY' ,'ADARE' ,'ASHBOURNE' ,'CARLOW TOWN' ,'CELBRIDGE' ,'CORK 1' ,'CORK CITY' ,
        'DROGHEDA' ,'GALWAY CITY' ,'KILKENNY CITY' ,'LIMERICK 1' ,'LIMERICK CITY' ,'MAYNOOTH' ,'NAAS' ,'NAVAN' ,'SWORDS' ,
        'WATERFORD CITY' ,'WICKLOW TOWN')   ) THEN 2
        ELSE [region_group]
    END

            END
    END AS AT_RiskLevelNew
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.9: SQL - Argument 09

A.10 SQL code - Argument 10

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) < (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) THEN 1
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 3.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) THEN 2
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.0) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) THEN 3
        WHEN ((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 4.5) <= CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))
            AND (CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])) <= (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0) THEN 4
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) > (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0)
            AND [primoccup_group] IN ('1','2','3') THEN 4
        WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) > (CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 5.0)
            AND [primoccup_group] IN ('4','5') THEN 5
    END AS AT_RiskLevelNew
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.10: SQL - Argument 10

A.11 SQL code - Argument 11

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) <= 35 THEN 1
        WHEN (CONVERT(NUMERIC(10,2),[LOAN_TERM]) > 35)
            AND CONVERT(NUMERIC(10,2),[LOAN_TERM]) <= 40 THEN 4
        WHEN CONVERT(NUMERIC(10,2),[LOAN_TERM]) > 40 THEN 5
    END AS AT_RiskLevelNew
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
```

Figure A.11: SQL - Argument 11

A.12 SQL code - Argument 12

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE WHEN Age_Plus_LoanTerm < 20 THEN 5
        WHEN Age_Plus_LoanTerm >= 20 AND Age_Plus_LoanTerm <= 45 THEN 1
        WHEN Age_Plus_LoanTerm > 45 AND Age_Plus_LoanTerm <= 55 THEN 2
        WHEN Age_Plus_LoanTerm > 55 AND Age_Plus_LoanTerm <= 65 THEN 3
        WHEN Age_Plus_LoanTerm > 65 AND Age_Plus_LoanTerm <= 75 THEN 4
        WHEN Age_Plus_LoanTerm > 75 THEN 5
    END AS AT_RiskLevelNew
FROM
(
SELECT
    [AGREEMENT_NO]
    ,[PRIMARY_DATE_OF_BIRTH]
    ,[COMPLETION_DATE]
    ,[LOAN_TERM]
    ,(DATEDIFF(YEAR, CONVERT(DATE,[COMPLETION_DATE]) , (CONVERT(DATE,[PRIMARY_DATE_OF_BIRTH])) ) *-1 ) AS AgeAtCompletion
    ,((DATEDIFF(YEAR, CONVERT(DATE,[COMPLETION_DATE]) , (CONVERT(DATE,[PRIMARY_DATE_OF_BIRTH])) ) *-1 )
     + CONVERT(NUMERIC(10,2),[LOAN_TERM])) AS Age_Plus_LoanTerm
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) Table1
```

Figure A.12: SQL - Argument 12

A.13 SQL code - Argument 13

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN RepaymentYearly *2 < NDI_Yearly THEN 1
        WHEN RepaymentYearly *2 > NDI_Yearly
            AND NDI_Yearly >= RepaymentYearly *1.6 THEN 2
        WHEN RepaymentYearly *1.6 > NDI_Yearly
            AND NDI_Yearly >= RepaymentYearly *1.2 THEN 3
        WHEN RepaymentYearly *1.2 > NDI_Yearly
            AND NDI_Yearly >= RepaymentYearly *0.5 THEN 4
        WHEN RepaymentYearly *0.5 > NDI_Yearly THEN 5
        WHEN (NDI_Yearly = 0) THEN 5
    END AT_RiskLevelNew
FROM
(
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 30000 THEN 0
        WHEN 30000 <= CONVERT(NUMERIC(10,2),[Joint_Total_Income])
        AND  CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 40000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.3),2)
        WHEN 40000 <= CONVERT(NUMERIC(10,2),[Joint_Total_Income])
        AND  CONVERT(NUMERIC(10,2),[Joint_Total_Income]) < 60000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.4),2)
        WHEN CONVERT(NUMERIC(10,2),[Joint_Total_Income]) >= 60000 THEN ROUND((CONVERT(NUMERIC(10,2),[Joint_Total_Income]) * 0.5),2)
    END AS NDI_Yearly
    ,ROUND(((CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))*(POWER(1.02,CONVERT(NUMERIC(10,2),[LOAN_TERM])))),2) AS PaybackAmount
    ,ROUND((ROUND(((CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]))*(POWER(1.02,CONVERT(NUMERIC(10,2),[LOAN_TERM])))),2)
     / CONVERT(NUMERIC(10,2),[LOAN_TERM])),2) AS RepaymentYearly
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) TableA
```

Figure A.13: SQL - Argument 13

A.14 SQL code - Argument 14

```sql
SELECT
    [AGREEMENT_NO]
    ,CASE
        WHEN [PP_DivBy_AD] <= 0.95 THEN 5
        WHEN [PP_DivBy_AD] > 0.95 AND
            [PP_DivBy_AD] <= 1.15 THEN 4
        WHEN [PP_DivBy_AD] > 1.15 AND
            [PP_DivBy_AD] <= 1.25 THEN 3
        WHEN [PP_DivBy_AD] > 1.25 AND
            [PP_DivBy_AD] <= 1.4 THEN 2
        WHEN [PP_DivBy_AD] > 1.4 THEN 1
    END AS AT_RiskLevelNew
FROM
(
SELECT
    [AGREEMENT_NO]
    ,[LOAN_PURPOSE]
    ,CASE WHEN [LOAN_PURPOSE] LIKE '%First Time Buyer%' THEN 1 ELSE 0 END AS FirstTimeBuyerFlag
    ,[AMOUNT_DRAWN]
    ,[PURCHASE_PRICE]
    ,CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) AS [AMOUNT_DRAWN_NUM]
    ,CONVERT(NUMERIC(10,2),[PURCHASE_PRICE]) AS [PURCHASE_PRICE_NUM]
    ,CONVERT(NUMERIC(10,2),[PURCHASE_PRICE]) / (CASE WHEN CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN])=0 THEN 1
     ELSE CONVERT(NUMERIC(10,2),[AMOUNT_DRAWN]) END) AS [PP_DivBy_AD]
FROM
    [HenrikDB].[dbo].[v1stDataSet_CleansedAdj]
) Table1
```

Figure A.14: SQL - Argument 14

Appendix B

Additional content - Arrears status codes

B.1 Arrears status codes

Code	**Arrears description**	**(Non-)default**
1	File passed to legal department	Default
2	Solicitors instructed	Default
3	Solicitors letter issued	Default
4	Solicitor applies for hearing date	Default
5	Hearing date obtained	Default
6	Order obtained / Ligation with arrangement	Default
7	Order obtained / No arrangement	Default
8	Transferred in judges list	Default
9	Possession order with stay	Default

Table B.1: Arrears status codes I

Code	**Arrears description**	**(Non-)default**
A	Arrangement in place	Non-default
B	No arrangement in place	Non-default
C	Live and non-performing	Default
D	Legal with arrangement	Default
E	Fraud	Default
F	Legal no arrangement	Default
G	Rebate received (no longer used for arrears management)	Non-default
H	Pre-legal with arrangement	Default
I	Indemnity claim pending	Default
J	Pre-legal no arrangement	Default
K	Credit transfer (no longer used for arrears management)	Non-default
L	Balance after termination	Default
M	Credit union (no longer used for arrears management)	Non-default

Table B.2: Arrears status codes II

Code	**Arrears description**	**(Non-)default**
N	Moratorium	Non-default
O	Order obtained for DSS benefit	Default
P	In possession	Default
Q	Moratorium interest only	Default
R	Repossessed and non-performing	Default
S	Awaiting deed to proceed	Default
T	Technical	Non-default
U	Unlikely to pay	Default
V	Secondary rental issue	Default
W	Agreement terminated	Default
X	Borrower deceased	Non-default
Y	Balance for write off	Default
Z	Shortfall customer services	Non-default

Table B.3: Arrears status codes III

Appendix C

Additional content - ICB codes

C.1 ICB codes

Code	**Code description**	**Can be succeeded by**
A	Arrangement Assigned to another Party	C,W,L
B	Borrower Cannot be Located	All
C	Completed Account	Nothing
D	Account in Dispute	All
F	Judgement Satisfied by Data Subject	Nothing
G	Goods in Merchantable Dispute	All
J	Judgement against Data Subject Obtained	F
K	Revoked Credit Card / Legal Credit Card / Revoked	A,K,C,W,L
L	Account settled for less than full amount	L

Table C.1: ICB codes I

Code	**Code description**	**Can be succeeded by**
M	Moratorium	M
N	Non Active Account	N
P	Pending Litigation	All
R	Repossession of Goods	C,W,L
S	Surrender of Goods	C,W,L
T	Terms Revised	Nothing
W	Written off Account	C,A
Z	No further updates available	Nothing
-	No history reported for this period	All

Table C.2: ICB codes II

Appendix D

DQ report preparation

D.1 DQ - SQL code to sample continuous features

```
SELECT TOP 500
--   A.[AGREEMENT_NO]
    B.[CASE_ID]
    ,CONVERT(INT,A.[primoccup_group]) AS [primoccup_group]
    ,CONVERT(INT,A.[region_group]) AS [region_group]
    ,CONVERT(MONEY,A.[Joint_Total_Income]) AS [Joint_Total_Income]
    ,CONVERT(MONEY,A.[AMOUNT_DRAWN]) AS [AMOUNT_DRAWN]
    ,CONVERT(NUMERIC(5,2),A.[LOAN_TERM]) AS [LOAN_TERM]
    ,CONVERT(DATE,A.[PRIMARY_DATE_OF_BIRTH]) AS [PRIMARY_DATE_OF_BIRTH]
    ,CONVERT(DATE,A.[COMPLETION_DATE]) AS [COMPLETION_DATE]
    ,CONVERT(DATE,A.[FIRST_VAL_DATE]) AS [FIRST_VAL_DATE]
    ,CONVERT(MONEY,A.[PURCHASE_PRICE]) AS [PURCHASE_PRICE]
FROM
    [HenrikDB].[dbo].[2ndDataSet_Cleansed] A
JOIN
    [HenrikDB].[dbo].[2ndDataSet_Map] B
ON
    A.[AGREEMENT_NO] = B.[AGREEMENT_NO]
ORDER BY A.[AGREEMENT_NO] DESC
```

Figure D.1: SQL code for extracting sample data to perform a DQ report on continuous features

D.2 DQ - R code to analyse continuous features

```
>
> library(readr)
  DQinputCont20170620 <- read_csv("C:/DIT/Thesis/DQinputCont20170620.csv",
                                  col_types = cols(AMOUNT_DRAWN = col_number
  (),
                                                   COMPLETION_DATE = col_date
  (format = "%Y-%m-%d"),
                                                   FIRST_VAL_DATE = col_date
  (format = "%Y-%m-%d"),
                                                   Joint_Total_Income =
  col_number(),
                                                   LOAN_TERM = col_number(),
                                                   PRIMARY_DATE_OF_BIRTH =
  col_date(format = "%Y-%m-%d"),
                                                   PURCHASE_PRICE = col_number
  ()))
  View(DQinputCont20170620)
  summary(DQinputCont20170620)
```

Figure D.2: R code for sample data DQ analysis - continuous features

D.3 DQ - SQL code to sample categorical features

```
SELECT TOP 500
--    A.[AGREEMENT_NO]
    B.[CASE_ID]
    ,A.[PRIMARY_OCCUPATION]
    ,A.[REGION_NAME]
    ,A.[LOAN_PURPOSE]
    ,A.[PRIMARY_SELF_EMPLOYED]
FROM
    [HenrikDB].[dbo].[2ndDataSet_Cleansed] A
JOIN
    [HenrikDB].[dbo].[2ndDataSet_Map] B
ON
    A.[AGREEMENT_NO] = B.[AGREEMENT_NO]
ORDER BY A.[AGREEMENT_NO] DESC
```

Figure D.3: SQL code for extracting sample data to perform a DQ report on categorical features